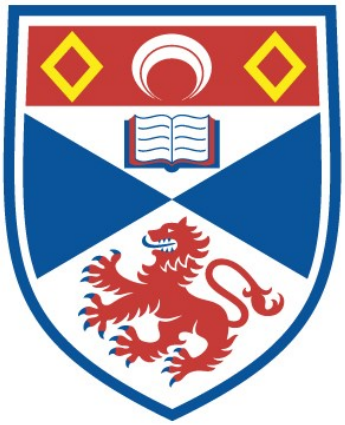
NONLINEAR MAGNETIC RECONNECTION

A. M. Colin

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



1987

NONLINEAR MAGNETIC RECONNECTION

A. M. COLIN

Thesis submitted for the Degree of Doctor of Philosophy

of the University of St Andrews

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

ABSTRACT

In many astrophysical problems magnetic reconnection plays a major role. Despite this reconnection theory remains incompletely understood, partly due to the strong non-linearity of the governing equations and the resulting difficulties in demonstrating analytical solutions. This thesis examines some fundamental aspects of reconnection theory: namely, the dynamics of driven and spontaneously reconnecting systems.

We first consider the dynamics of a driven reconnecting system by numerically modelling a configuration consisting of two oppositely oriented flux systems with a variety of different boundary conditions and internal parameters. The results indicate that the rate of reconnection is chiefly dependent on the magnetic Reynolds number, but that the plasma flow is weakly dependent on this parameter, being more affected by the curvature of incoming magnetic field. Scaling laws for the dimensions of the diffusion region are derived, and the existence of several reconnection regimes is shown.

Using the same computer code we also simulate tearing modes in Cartesian geometry under different boundary conditions. By imposing a suitable perturbation a magnetic island is generated. The plasma flows show marked differences for the different boundary conditions implemented.

Lastly, we examine some aspects of the coalescence instability. The usual flux function taken to represent a tearing mode island in the linear growth phase is shown to be erroneous, and we derive a correct expression. We show that under certain conditions there exists a threshold to coalescence that depends on the island wavenumbers and the associated perturbation.

ACKNOWLEDGEMENTS

It is a great pleasure to be able to thank Professor Eric Priest for his supervision of my research studentship over the last three years at St Andrews, Dr Alan Sykes and the staff of the Theory Division at Culham Laboratory for help and guidance, Dr Alan Hood for his supervision of Chapter 5, Dr Alastair Robertson for much discussion and advice on numerical simulation, Culham Laboratory for a generous allocation of computer time, colleagues at St Andrews, and friends and family for their unflagging support and encouragement.

I am grateful to the Science and Engineering Research Council and Culham Laboratory for financial support.

DECLARATION

I, Andrew Mark Colin, declare that the following thesis is a record of research work carried out by me, that the thesis is my own composition, and that it has not been submitted in any previous application for a higher degree.

Signature of Candidat

Date 22nd May 1987

CERTIFICATE

I certify that Andrew Mark Colin has satisfied the conditions of the Ordinance and Regulations and is thus qualified to submit the accompanying application for the degree of Doctor of Philosophy.

Signature of Supervisor ....

Date ....22.5.87.....

POSTGRADUATE CAREER

I was admitted into the University of St Andrews as a research student under Ordinance General No. 12 in October 1983 and as a candidate for the degree of Ph.D. in October, 1984, to work on aspects of Magnetic Reconnection under the supervision of Professor E.R. Priest.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction to magnetic reconnection

Magnetic reconnection is a phenomenon of fundamental importance in astrophysical and laboratory plasmas. A magnetic field occupying the same spatial region as a perfectly conducting plasma or fluid is tied to the fluid and convected along with the fluid motions, so the initial topology is preserved in time. However, the introduction of even a small amount of resistivity allows oppositely oriented magnetic field lines to come together at current sheets (where a component of magnetic field changes sign) and break, or reconnect, into a different configuration. Although the width of such current sheets may be very small compared with the scale length of the plasma, this localised effect alters the global topology of the field. Magnetic reconnection may therefore act as a decoupling mechanism; a topological constraint on the plasma is removed and new, lower energy states are accessible.

This has two major effects on the plasma. Firstly, new field topologies may be created, which allow transport of plasma ions and impurities into previously inaccessible regions. Secondly, magnetic energy within the magnetic field may be released and converted into kinetic and heat energy. Both phenomena are of considerable interest to astrophysicists and laboratory plasma physicists.

As an illustration of the importance of reconnection we give three examples in the context of astrophysical, geophysical and laboratory plasmas:

(i) In a *solar flare*, reconnection is believed to be the only mechanism that can give rise to the observed release of very large amounts of energy (up to $\simeq 10^{25}$ J) over the observed short time-scales ($\simeq 10^{3}$ s). In a two-ribbon flare, a magnetic arcade passes through a series of magnetohydrostatic equilibria, until some critical shear in the field is reached. The system then becomes unstable, begins to reconnect and erupts outwards into the corona to produce loops and Hα ribbons, together with a wide range of electromagnetic radiation, an interplanetary blast wave, relativistic and nonrelativistic nuclei, and high-energy electrons.

(ii) Magnetic reconnection is involved in the interaction of the Earth's *magnetosphere* and the solar wind. A current sheet is formed when the interplanetary magnetic field meets the Earth's magnetic field. The field lines then reconnect across the sheet, allowing protons and helium ions from the solar wind to enter the dayside magnetosphere. Evidence for reconnection has been provided by the ISEE spacecraft in the form of both quasi-steady processes and flux transfer events (Bagenal, 1985). Similar phenomena have been observed in the Jovian magnetosphere by the Voyager 1 and 2 spacecraft (Nishida, 1984)

(iii) Magnetic reconnection is an important cause of *disruptions* in tokamaks and activity in reverse field pinches. Temperatures of around one hundred million degrees Kelvin are required for the successful attainment of the Lawson criterion, which relates the number density and confinement time for a plasma of deuterium nuclei to reach the conditions under which a fusion reaction may occur. To achieve such conditions in the laboratory requires that the plasma be magnetically confined, ususally in the form of a torus. With a suitable containment

vessel and strong axial magnetic field the torus may be made stable to ideal instabilities. However, there still remain the resistive instabilities (such as the tearing mode) which can degrade the confinement properties of the plasma, enhance anomalous transport of impurities, and even cause major disruptions that damage the containment vessel.

Despite the fundamental importance of reconnection theory, it is still incompletely understood. The subject has traditionally been divided into the two schools of *spontaneous* and *driven* reconnection, with each having its own (largely independent) literature.

Perhaps the best-known spontaneously reconnecting magnetohydrodynamical phenomenon is the tearing mode instability, in which the initially anti-parallel field lines about a current sheet 'tear' and reconnect to form a chain of magnetic islands. This process is highly time-dependent (Furth, Killeen and Rosenbluth, 1963). By contrast, theories of driven reconnection have tended to concentrate on steady-state solutions to the MHD equations, rather than on time-dependent phenomena. Thus results from the two schools have not been directly applicable, even though the same equations and physical assumptions are usually used.

It has only been over the last decade, mostly through the medium of large-scale computer simulation, that any further progress has been made in the understanding of reconnection theory, and one of the most striking results is how closely these two schools are now related. For instance, consider the coalescence instability, in which an ideal instability forces together two or more magnetic islands, possibly in a nonlinear phase of tearing. This phenomenon will be governed by a

mixture of ideal and resistive MHD effects. Although this instability occurs spontaneously, the rate at which the islands coalesce will be determined by the maximum rate at which their field lines can reconnect and is thus determined by the dynamics of driven reconnection. Indeed, we may regard any spontaneously reconnecting system as an ideal system with the dynamics being locally controlled by driven systems, or a driven system as a small cross-section of a spontaneously reconnecting system. A more fruitful approach to use than this rather arbitrary division is to assume a set of equations that model the desired physical system and to vary the boundary and initial conditions under which they are solved. The bulk of the research presented below forms such a study.

1.2 Outline of thesis

The remainder of Chapter 1 reviews the current state of analytical reconnection theory and describes the most important work published to date on numerical reconnection simulations. In Chapter 2 we describe the fundamental algorithms employed in the computer code SHASTA, together with their implementation, and give a brief account of recent research on the possiblities and limitations of numerical simulation. Chapter 3 shows how we adapted the code to simulate two-dimensional driven reconnection, and found a series of steady states when the inflow field line curvature, magnetic Reynolds number and inflow speed were varied. The results are compared with the recent study by Biskamp (1986) and scaling laws for the reconnection rate and dimensions of the internal region were deduced to be compared with analytical theory. Chapter 4 is an account of how the same code was used to model tearing modes with two sets of boundary conditions. We compared various plasma parameters for

the two cases and studied the reconnection rate as functions of time. Chapter 5 has an analytical study of the ideal coalescence instability for a magnetic configuration generated by tearing mode instability, as the field profile due to Finn and Kaw (1977) used in all previous studies is shown to be unrealistic. Chapter 6 concludes with a summary of the major results and some suggestions for further work.

1.3 The MHD equations

Throughout the next three chapters of this thesis we shall be studying solutions to the full non-linear, compressible, resistive, two-dimensional magnetohydrodynamical equations:

$$\frac{\partial \underline{B}}{\partial t} = \underline{\nabla} \times (\underline{v} \times \underline{B}) + \eta \nabla^2 \underline{B} \qquad (1.3.1)$$

Induction

$$\frac{\partial \rho}{\partial t} + \underline{\nabla} \cdot (\rho \underline{v}) = 0 \qquad (1.3.2)$$

Mass continuity

$$\rho \left(\frac{\partial \underline{v}}{\partial t} + (\underline{v} \cdot \underline{\nabla}) \underline{v} \right) = \underline{j} \times \underline{B} - \underline{\nabla} p \qquad (1.3.3)$$

Momentum

$$\frac{\partial p}{\partial t} + (\underline{v} \cdot \underline{\nabla}) p = - \gamma p \, \underline{\nabla} \cdot \underline{v} \qquad (1.3.4)$$

Energy

where

$$\underline{j} = \underline{\nabla} \times \underline{B} \qquad (1.3.5)$$

together with an Ohm's law

$$\eta \underline{j} = \underline{E} + \underline{v} \times \underline{B} \tag{1.3.6}$$

and Maxwell's equation

$$\nabla \cdot \underline{B} = 0 \tag{1.3.7}$$

These are supplemented by the equation of state for an ideal gas

$$p = \rho T \tag{1.3.8}$$

where the quantities B, v, ρ, p, X, t, E, T are non-dimensionalised as follows:

$$\underline{B} = \underline{B}' / B_0' \qquad \rho = \rho' / \rho_0' \qquad \underline{v} = \underline{v}' / v_a'$$

$$p = \frac{2\mu_0 p'}{B_0'^2} \qquad \underline{j} = \frac{\underline{j}' \mu_0 L_0'}{B_0'} \qquad v_a' = \frac{B_0'}{(\mu_0 \rho_0)^{\frac{1}{2}}}$$

$$\underline{x} = x_0' / L_0' \qquad t = t' v_a' / L_0' \qquad \underline{E} = \underline{E}' / v_a' B_0'$$

$$T = R \rho_0' T' / p'$$

where B', v', ρ', p', j', X', t' ,E', T' are the dimensional magnetic field, flow velocity, mass density, pressure, current density, spatial coordinate, time, electric field and temperature, respectively. B_0' and ρ_0' are the values of the magnetic field and plasma density in the external

region, and L_o' is the size of the computational domain. γ is the ratio of specific heats (= 5/3), R is the gas constant, μ_o is the free space magnetic permeability and η is the dimensionless magnetic resistivity. η is related to the dimensional magnetic resistivity η', by

$$\eta = \eta' / L_o' V_A' \qquad (1.3.8)$$

which is the inverse of the Lundquist number, more commonly referred to as the magnetic Reynolds number. We adopt this usage throughout the rest of the thesis.

The reader should note that the standard MHD approximations have been made when combining the Euler fluid equations and Maxwell's equations of electromagnetism to derive the magnetohydrodynamical equations:

(i) The displacement current $\frac{\partial \underline{D}}{\partial t}$ is negligible compared to the conduction current $\underline{J}$.

(ii) The plasma is electrically neutral, so that $\rho_c \ll \rho_n$ where ρ_n and ρ_c are the densities of neutral and charged ions within the plasma. For a fuller account of the approximations see, for instance, Priest (1982, p 92). Note also that our form of Ohm's law is very simple compared to the more generally used expression

$$\underline{E} + \underline{v} \times \underline{B} = \eta \underline{j} + \frac{1}{en} (\underline{j} \times \underline{B} - \nabla p_e) \qquad (1.3.9)$$

(Chen, 1974, p 164) where e is the charge on the electron, n is the number density of charged particles and p_e is the electron pressure. In particular, we have neglected the Hall current term.

1.4 Analytical models of reconnection

In this section we give an overview of the most important analytical MHD reconnection models. We follow the historical development of the subject by considering (in order) the Sweet-Parker, Petschek and Priest-Forbes models.

Sweet (1958) and Parker (1963) considered a two-dimensional, steady-state system with antiparallel inflow magnetic fields on either side of a neutral line (Figure 1.1). They argue that in a steady state the speed V_i at which plasma carries magnetic field into a sheet of thickness is just the speed η/ℓ at which the magnetic field diffuses through the plasma:

$$V_i = \eta / \ell \qquad (1.4.1)$$

where η is the magnetic resistivity. For an incompressible medium the current sheet accelerates the outgoing plasma to the Alfven speed

$$V_a = B_i / (\mu_0 \rho)^{\frac{1}{2}} \qquad (1.4.2)$$

so we may use mass conservation to equate the inflow and outflow mass fluxes:

$$L V_i = \ell V_a \qquad (1.4.3)$$

where L is the length of the sheet. This assumes that the current sheet retains a width ℓ over the length of the sheet L, and that L is in fact the scale length of the system.

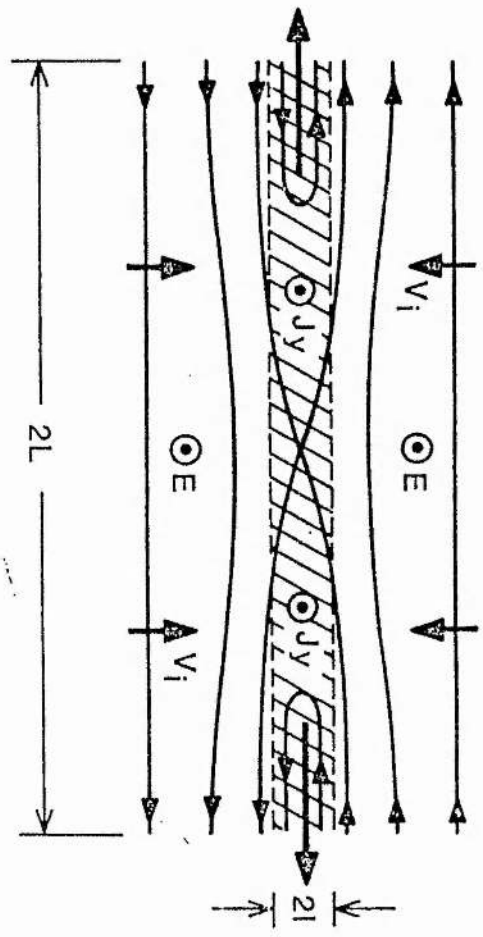


Figure I.I Reconnecting field lines in the Sweet-Parker model

Eliminating ℓ between (1.4.1) and (1.4.3) gives the Alfven Mach number

$$M_i = v_i / v_a \qquad (1.4.4)$$

from which we may derive the rate at which field lines can be carried into the system. For a steady state this represents a measure of the reconnection rate. We find

$$M_i = R_m^{-\frac{1}{2}} \qquad (1.4.5)$$

where the magnetic Reynolds number is

$$R_m = \frac{L_0 v_a}{\eta} \qquad (1.4.6)$$

The difficulty with this model is that energy is released from the magnetic field at rates which are far too slow to account for flare observations. Taking a magnetic Reynolds number of $\simeq 10^{8}$ to 10^{10}, an Alfven speed of $\simeq$ 100 km/s and a scale length of $\simeq 10^{4}$m gives a timescale for energy release of tens of days, which is clearly inadequate when compared to the observed timescales of several tens of minutes. To give more accurate results a very much smaller length scale must be taken, which is equivalent to making the inflow speed comparable to the ambient Alfven speed.

The difficulty of reconection rates which are far too small was surmounted by Petschek (1964) in a classic paper based on an idea which he credits to A.R. Kantrowitz. Petschek suggested that the current sheet

may bifurcate into two pairs of slow magnetohydrodynamical shock waves which propagate from a central current sheet. The essential difference between this model and that of Sweet and Parker is that the central diffusion region may be very small, thus decreasing R_m and allowing much higher inflow speeds. The shock waves stand in the flow and (in the compressible case) convert magnetic energy to the thermal and kinetic energy of two hot jets, which are accelerated from the diffusion region to reach the local Alfven speed (Figure 1.2).

Soward and Priest (1977) have derived an expression for the reconnection rate by finding incompressible, two-dimensional solutions to the MHD equations which are asymptotic and hence valid at large distances from the diffusion region. For large R_m they find a maximum reconnection rate of

$$M_e = \frac{\pi}{8 \log_e R_m} \qquad (1.4.7)$$

(where M_e is the external Mach number) which in practice lies between 0.01 and 0.1 and is very much greater than the Sweet-Parker value.

Sonnerup (1970) has proposed a model which, while not directly physically applicable, nevertheless has features in common with several numerical simulations. He proposes a model with four slow mode expansion waves which are generated at corners in the inflow. The model predicts that for this configuration the Mach number in the inflow region may takes values up to unity.

Vasyliunas (1975) has performed a detailed mathematical analysis of the Petschek and Sonnerup mechanisms. As well as putting the Petschek mechanism onto a firm mathematical basis, he observes that the

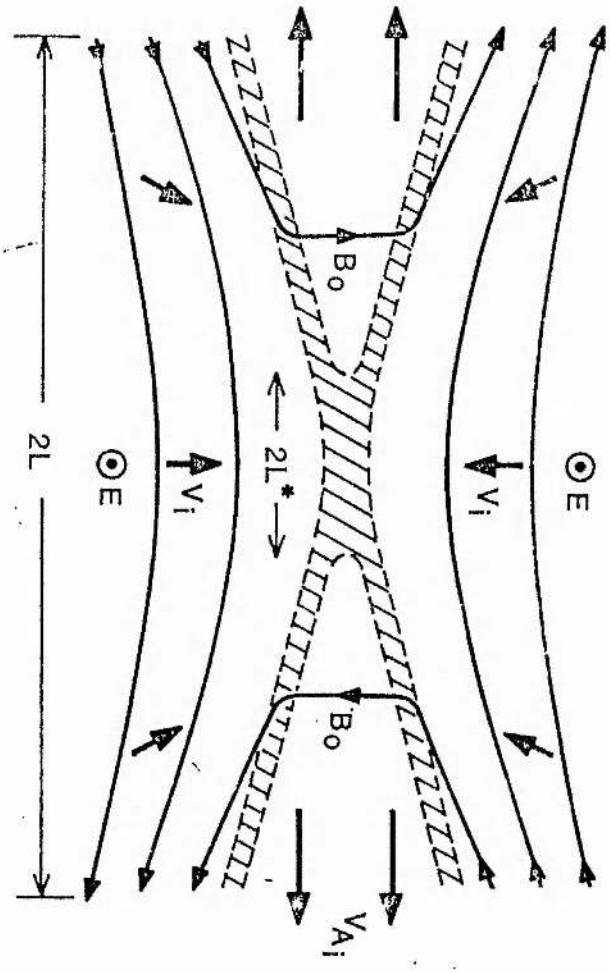


Figure I.2 Reconnecting field lines in the Petschek model

streamlines for a Petschek solution converge on approaching the diffusion region, an effect caused by a lessened pressure along a line from the diffusion region outwards. Noting also that the magnetic field decreases on approaching the current sheet, the solution is an example of a fast mode expansion. For the Sonnerup solution the streamlines diverge and the magnetic field increases on approaching the current sheet, thus forming a slow mode expansion.

Priest and Forbes (1986) have presented a new series of analytical solutions that include the Petschek solution and a Sonnerup-like solution as special cases. They take a finite two-dimensional region ($|x| \leq 1$, $|y| \leq 1$) and linearise about a uniform horizontal field $B_0\underline{x}$ in the upper part of the region ($y \geq 0$) by setting

$$\underline{B} = B_0\, \underline{x} + \underline{B}_1 + \ldots \quad (1.4.8)$$

$$\underline{V} = V_{1Y}\, \underline{y} + \underline{V}_2 + \ldots \quad (1.4.9)$$

where $V_{1Y} = M_0\, V_{A0}$, and where the lower half of the box consists of field lines oppositely oriented from those in the upper half. The boundary conditions for the solution in the upper half are:

$B_{1x} = 0$ on $y = 1$

$= 0$ on the sides ($|x| = 1$)

$B_{1Y} = f(x)$ on $y = 0$

where $f(x)$ is a function that gives the appropriate normal magnetic field generated by the slow mode shock waves. The solution is

$$B_{1x} = \sum_{n=0}^{\infty} a_n [5 - \cos(n\pi x)] \sin[n\pi(1-y)] \quad (1.4.10)$$

$$B_{iy} = \sum_{n=0}^{\infty} a_n \sin(m\pi x) \cosh[m\pi(1-y)] \qquad (1.4.11)$$

where m = n + ½ and

$$a_n = \frac{4 B_N \sin(m\pi L/Le)}{(L/Le) m^2 \pi^2 \cosh(m\pi)} \qquad (1.4.12)$$

The parameter b categorises the different types of solutions. b = 0 gives a Petschek solution; b = 1 gives a Sonnerup-type solution. When b < 0 we have a family of slow-mode compressions with strongly convergent flows, while with b > 1 there exists a flux pile-up regime of slow-mode expansions, with strongly divergent flow and an increase in magnetic field on approaching the current sheet. A hybrid regime exists when 0 < b < 1, for which there is a fast-mode expansion on the y-axis and a slow-mode expansion on the line |x| = 1. Work is currently in progress (Robertson and Priest, 1987) to provide a numerical counterpart to these solutions.

It is of interest to derive some estimates for the inner (at the current sheet) and external (at the entrance to the inflow region) Alfven Mach numbers. For reference the results from the Sweet-Parker and Petschek theories are grouped together in table 1.1.

From equation (44) of Priest and Forbes (1986) we have

$$\frac{B_e^2}{B_i^2} = \frac{M_i}{M_e} \qquad (1.4.13)$$

where B_e, B_i, M_e, M_i are the external and internal magnetic fields, external and internal Mach numbers, respectively. Equation (4) of the

same paper relates the internal, external and normal magnetic field at the neutral line:

$$B_i = B_e - \frac{4 B_N}{\pi} \log_e \left(\frac{L_e}{L} \right) \qquad (1.4.14)$$

Since

$$\frac{B_N}{B_e} = \frac{V_e}{B_e} (\mu \rho)^{\frac{1}{2}} = M_e \qquad (1.4.15)$$

we may combine these results and substitute back into (1.4.13) to find the internal Mach number for the Petschek model in terms of the external Mach number.

Table 1.1 Internal and external Alfven Mach numbers

Sweet-Parker reconnection:

$$M_\infty = R_m^{-\frac{1}{2}}$$

$$M_i = R_m^{-\frac{1}{2}}$$

Petschek reconnection:

$$M_\infty = \pi / 8 \log_e R_m$$

$$M_i = M_e \left(\frac{B_e}{B_i} \right)^2 = M_e \left(1 - \frac{4 M_e}{\pi} \log_e \left(\frac{L_e}{L} \right) \right)^{-2}$$

These results will be required in Chapter 3 so that we may compare results from the numerical work and the analytical theory.

1.5 Previous numerical work

This section presents a brief overview of previously published work on numerical reconnection experiments that are similar to our work.

The first work on numerical reconnnection was performed by Fukao and Tsuda (1973) who solved the incompressible time-dependent equations of motion and induction over a rectangular region with boundary conditions imposed so as to give stagnation-point flow. Using a Friedrichs-Lax scheme and an effective magnetic Reynolds number of 50-500, they produced a non-zero component of magnetic field perpendicular to the initial configuration, thus demonstrating that reconnection was occurring. The experiment had the limitations of insufficient resolution to resolve the diffusion region, and of not being run for long enough to allow a steady state to develop.

Ugai and Tsuda (1977) study a similar configuration, using a two-step Lax-Wendroff scheme to solve the compressible MHD equations with an energy equation including ohmic heating; an artificial viscosity is imposed to ensure stability of the numerical scheme. They take an initial configuration of a one-dimensional current sheet and assume that R_m (= 1000 in the external region) is diminished by a factor of 100 within a circle of given radius about the origin. This generates an x-point in the magnetic field at the origin and produces Petschek-type flows, with a plasma jet directed along the x-axis. An integration over the box shows that the magnetic energy in the box decreases over the duration of the experiment despite a net inflow of magnetic energy; the

reconnection mechanism is providing conversion into thermal and kinetic energy. The same authors (Tsuda and Ugai 1977, Ugai and Tsuda 1979a, b) have presented further runs of the same experiment, in which they show that holding the electric field vB constant on the inflow boundary gives evolution to a steady state. The effect of increasing the plasma resistivity is to shrink the diffusion region and to change the reconnection rate, which is found to be in agreement with Soward and Priest's 1977 result.

Ugai (1981 , 1982, 1983) continues this work by considering the effect of different boundary conditions. He finds that replacing the free inflow boundary by a rigid wall over which there is no mass or field flux only changes the reconnection rate slightly, but that replacing the exit boundary by the same type of wall inhibits fast reconnection. Instead, an o-point forms where previously an x-point had been observed.

Hayashi and Sato (1978) solve a similar set of equations but obtain driven reconnection by imposing the flow speed on their inflow boundary, rather than letting it free float. They assume zero magnetic resistivity (ideal MHD) everywhere except where the current density exceeds a critical value to model an anomalous diffusivity. With strongly curved incoming field lines and a free-floating exit boundary they state that a steady state is reached after 15-20 Alfven transit times, although their graphs of electric field at the x-point (a measure of the reconnection rate) do not confirm this. A Petschek-type flow is found, together with an associated bifurcation of the initial current sheet into two standing shock waves. A run of the same code for 'confined' reconnection (exit

boundary replaced by a rigid wall) shows that an island forms within the box, and that an O-point rather than an X-point forms at the origin.

Biskamp (1985, 1986) presents some interesting solutions to the incompressible equations at high resolution and magnetic Reynolds number, for which he also has a driven inflow and free floating exit boundary. Despite the appearance of Petschek-type flows and an associated current-sheet bifurcation, he interprets his results as being nearer a flux pile-up regime. The appearance of a reversed current region at the downstream end of his diffusion region suggests that a reconection regime of the type considered by Syrovatsky (1971) is taking place. Because of the importance of these simulations and their relevance to the study presented in this thesis, a fuller discussion is postponed to Chapter 3 where the results from the two sets of experiments are compared.

1.6 Reconnection regimes

We may summarise and extend Section 1.4 in a classification of six distinct reconnection regimes for an electrically conducting plasma, identified by inflow speed V_∞ and proposed by Priest (1985):

(I) Very slow reconnection occurs when a system passes through a series of quasi-potential states. For this to occur, the inflow speed V_∞ must be less than the global diffusion speed η/L_e, where V_∞ is the global ambient length scale.

(II) Slow reconnection, or the Sweet-Parker regime (1958, 1963) occurs when

$$\frac{\eta}{L_e} < V_e < \frac{V_a}{R_m^{1/2}}$$

where $V_a / R_m^{\frac{1}{2}}$ is the Sweet-Parker reconnection rate and Rm is the ambient magnetic Reynolds number. The diffusion region (or current sheet) has length Le and the magnetic energy is converted equally into heat and kinetic energy.

(III) Petschek-Sonnerup reconnection develops when

$$\frac{V_a}{R_m^{\frac{1}{2}}} < V_e < V_{max}$$

where V_{max} scales as $1/\log(R_m)$, a result first given by Petschek (1964). He suggests that a current sheet may bifurcate into four slow mode MHD shock waves propagating from a central region where the Sweet-Parker regime is dominant. Since the dimensions of this central current sheet will be much less than the global length scale, the magnetic Reynolds number governing the reconnection rate within the sheet will accordingly be lower and so the global reconnection rate will be much higher. In fact, this internal Rm may become of order one and so account for very rapid reconnection, which has made this model attractive to theorists.

(IV) Unsteady flux pile-up reconnection occurs when

$$V_{max} < V_m$$

so that a steady state cannot occur. This regime postulates the existence of a maximum reconnection rate; magnetic field is brought into the system faster than it can be reconnected and so the flux accumulates at the diffusion region, causing it to lengthen into a long current sheet. Eventually the system will resemble a Sweet-Parker current sheet,

save that the magnetic field strength will increase rather than decrease on approaching the current sheet.

(V) Impulsive bursty reconnection may be induced after flux pile-up occurs. The central current sheet develops to such a length that it becomes unstable to the tearing mode instability, leading to the formation of chains of magnetic islands. These may be expelled from the box in the plasma jet at the end of the current sheet, or coalesce together to form larger islands. Sakai (1983) identifies signatures in X-ray emission from coronal loops as a manifestation of this instability.

(VI) Turbulent reconnection may develop from the previous regime at sufficiently high magnetic Reynolds number, with current filamentation developing on progressivly lower length scales. However, the difficulties of modelling such a process analytically or computationally are very great, and the effect on the global reconnection rate of plasma turbulence is uncertain.

CHAPTER 2

NUMERICAL SOLUTION OF THE MAGNETOHYDRODYNAMICAL EQUATIONS

2.1 The MHD equations

Throughout the next three chapters of this thesis we shall be studying solutions to the full non-linear, compressible, resistive, two-dimensional magnetohydrodynamical equations given in Chapter 1 (1.3.1 - 1.3.4). To produce a numerical solution we must expand these equations out into components and cast them into the form of a generalised continuity equation:

$$\frac{\partial f}{\partial t} + \nabla \cdot (f \underline{v}) = S(f) \qquad (2.1.1)$$

where f is the transported quantity and S is a source term. We have

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v_x) + \frac{\partial}{\partial z}(\rho v_z) = 0 \qquad (2.1.2)$$

$$\frac{\partial}{\partial t}(\rho v_x) + \frac{\partial}{\partial x}(\rho v_x^2) + \frac{\partial}{\partial z}(\rho v_x v_z) = -\frac{\partial}{\partial x}(p + B_z^2/2) + B_z \frac{\partial B_x}{\partial z} \qquad (2.1.3)$$

$$\frac{\partial}{\partial t}(\rho v_z) + \frac{\partial}{\partial x}(\rho v_x v_z) + \frac{\partial}{\partial z}(\rho v_z^2) = -\frac{\partial}{\partial z}(p + \frac{B_x^2}{2}) + B_x \frac{\partial B_z}{\partial x} \qquad (2.1.4)$$

$$\frac{\partial B_x}{\partial t} + \frac{\partial}{\partial x}(B_x v_z) = \frac{\partial}{\partial z}(B_z v_x) - \frac{\partial}{\partial z}\eta\left(\frac{\partial B_x}{\partial t} - \frac{\partial B_z}{\partial x}\right) \qquad (2.1.5)$$

$$\frac{\partial B_z}{\partial t} + \frac{\partial}{\partial z}(B_z v_x) = \frac{\partial}{\partial x}(B_x v_z) - \frac{\partial}{\partial x}\eta\left(\frac{\partial B_z}{\partial x} - \frac{\partial B_x}{\partial z}\right) \qquad (2.1.6)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v_x) + \frac{\partial}{\partial z}(\rho v_z) = -(\gamma - 1)\rho\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z}\right) \qquad (2.1.7)$$

SHASTA solves the MHD equations by splitting them into advective and diffusive parts. This may be illustrated by examination of equation (2.1.1). Firstly, we solve the advective (hyperbolic) part of the equations, by setting the source term equal to zero:

$$\frac{\partial f}{\partial t} + \underline{\nabla} \cdot (f \underline{v}) = 0 \qquad (2.1.8)$$

Secondly, we use the transported value f' determined from the first part as initial conditions for the diffusive (parabolic) part of the equations. We set the divergence term equal to zero:

$$\frac{\partial f'}{\partial t} = S(f') \qquad (2.1.9)$$

The rest of this chapter is a description of some of the numerical methods used in our computer program that solves the above equations in two-dimensional Cartesian geometry. The program (or code) is referred to as SHASTA, after the particular flux-corrected transport algorithm used in the advective subroutines. For a fuller description the reader is referred to the original papers by Boris and Book (1973, 1976), Weber's thesis (1978) and the complete listing of the code given in Appendix B.

2.2 Computer simulation

Traditionally, the three methodologies of astronomy and astrophysics have been theory, experiment and observation. With the advent of widely distributed, powerful computer facilities over the last decade a fourth is now coming into widespread use; that of large-scale numerical simulation.

Computing the behaviour of a physical system by specifying the governing laws and advancing them in time is a technique that lies midway between the areas of the applied mathematician and the experimentalist. In most cases the performance of a calculation on a computer closely resembles the performance of a physical experiment, in that the analyst runs an experiment without necessarily knowing what the results will be. However, the results of a numerical experiment can be much easier to compare with analytical work than with corresponding observations in the laboratory or above the atmosphere. In a computational experiment the analyst has complete control over his model universe. He may completely specify the initial configuration and internal parameters; his experimental probes do not disturb the experiment; he can study a purely two-dimensional configuration. He is also able to try what neither the theoretician nor the experimentalist may, by testing the sensitivity of phenomena to independent theoretical approximations (Arnett, 1985).

Most serious work in numerical simulation has come to require the power of supercomputers, such as those built by Cray Research, which rely on vector processing architectures. However, we expect that in the

next decade a number of machines with non-Neumann architectures (Pearson, Richardson and Toussaint, 1985) will be specially built to treat problems of comparable complexity and processing cost to those examined in the subsequent chapters. These computers will be based on large arrays of microprocessors configured in a two- or three-dimensional matrix so that one processor is available for every gridpoint at which the governing equations are to be solved. It may be that such machines will eventually become more powerful for application to this class of problem than non-dedicated supercomputers, where constraints such as the time taken for an electrical signal to cross the computer at the speed of light will put a limit on the ultimate speed of operation.

The program that produced the results presented in Chapters 3 and 4 was developed on the two DEC VAX 11/785 machines at St Andrews, and was run both at St Andrews and on the UKAEA's PRIME at Culham Laboratory, Oxfordshire, as a 'front-end' machine for the Harwell CRAY-1.

2.3 The SHASTA algorithm

In this section we give a brief account of SHASTA, one of a class of flux-corrected transport (FCT) algorithms first presented by Boris (1973). These schemes are designed to perform accurate transport of discontinuities in fluid dynamical simulations, and hence the use of such an algorithm will be particularly helpful when we come to simulate MHD shocks in the Petschek model.

Conceptually, such an algorithm consists of two stages: a transport or convective stage, followed by an anti-diffusive or corrective stage. Both parts are conservative. Further, an FCT scheme conserves positivity

of actual mass and energy densities, so steep gradients and inviscid shocks are handled much more accurately than by a conventional second-order scheme such as the Lax-Wendroff (Roache, 1976).

As an illustration of the method we here give a simple geometrical interpretation of FCT applied to the one-dimensional continuity equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v) = 0 \qquad (2.3.1)$$

The method may easily be extended to multi-dimensional applications.

Stage I: Diffusive transport

Figure 2.1(a) shows an Eulerian, uniformly spaced grid at the beginning of an iteration (t = 0). We assume that the densities are known and that the velocities $\{v_j^{\frac{1}{2}}\}$ are known at t = δt/2, half a timestep ahead. We require the densities $\{\rho_j^{1}\}$ at the end of the timestep when t = δt.

The method begins by considering a typical fluid-element trapezoid, formed by connecting adjacent densities with straight line segments. The density profile thus constructed is consistent with the definition of the total mass

$$M_0 = \sum_j \rho_j \, \delta x \qquad (2.3.2)$$

On Figure 2.2(b) we have sketched short arrows, showing how these fluid elements and their boundaries might convect in a Lagrangian sense. To ensure positivity, Boris restricts his analysis to $| v \, \delta t/\delta x | < \frac{1}{2}$ where $v = \max(|v_j|)$ so that no fluid element boundary convects past a

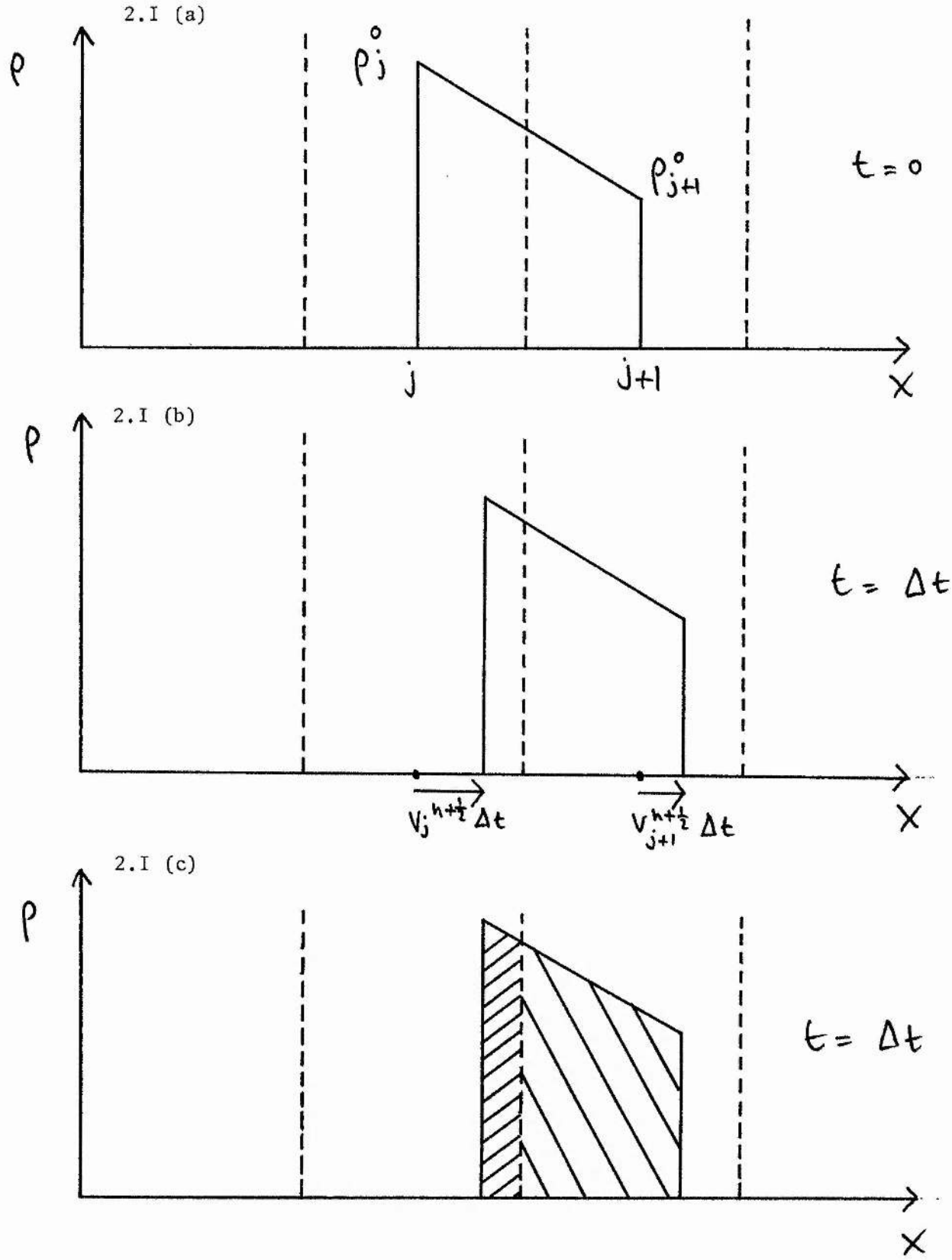


Figure 2.I Convected fluid trapezoid

cell boundary. In our full code we take a more restrictive condition that enables us to treat MHD waves; this Courant-Friedrichs-Lewy (or CFL) condition is described below.

We follow the motion of the fluid element for one timestep in a Lagrangian sense. The complete transport prescription, relating $\{\rho_j^{n+1}\}$ to $\{\rho_j^n\}$, is

$$\rho_j^{n+1} = \tfrac{1}{2}\Phi_-^2(\rho_{j-1}^n - \rho_j^n) + \tfrac{1}{2}\Phi_+^2(\rho_{j+1}^n - \rho_j^n) + (\Phi_+ + \Phi_-)\rho_j^n \qquad (2.3.3)$$

$$\Phi_\pm = \left(\tfrac{1}{2} \mp v_j^{\frac{1}{2}}\frac{\delta t}{\delta x}\right)\Big/\left[1 \pm (v_{j\pm1}^{\frac{1}{2}} - v_j^{\frac{1}{2}})\frac{\delta t}{\delta x}\right] \qquad (2.3.4)$$

where for greater accuracy the $\{\bar{v}_j^{\frac{1}{2}}\}$ derived from linear interpolation back on the grid may be used (Figure 2.2(c)). These are given by

$$\bar{v}_j^{\frac{1}{2}} = [1 - \xi_j/2]\, v_j^{\frac{1}{2}} + (\xi_j/2)\, v_{j+1}^{\frac{1}{2}} \qquad (v_j^{1/2} \geq 0) \qquad (2.3.5)$$

$$\bar{v}_j^{\frac{1}{2}} = [1 - \xi_j/2]\, v_j^{\frac{1}{2}} + (\xi_j/2)\, v_{j-1}^{\frac{1}{2}} \qquad (v_j^{1/2} < 0) \qquad (2.3.6)$$

where $\xi_j \equiv |v_j^{1/2}\, \delta t/\delta x|$.

For a uniform velocity field this reduces to the simpler form

$$\rho_j^{n+1} = \rho_j^n - \frac{\xi}{2}(\rho_{j+1}^n - \rho_{j-1}^n) + \left(\frac{1}{8} + \frac{\xi^2}{2}\right)(\rho_{j+1}^n - 2\rho_j^n + \rho_{j-1}^n) \qquad (2.3.7)$$

This formula is a simple two-sided differencing of the convection term, plus a strong zero-order diffusion. Without this velocity-independent diffusion we just have the result found by using the two-step Lax-Wendroff algorithm (Mitchell, 1969; Roache, 1976), with the usual second-order dispersion and velocity-dependent diffusion.

The strong diffusion corresponds to acting on the initial density profile $\{\rho_j^0\}$ in the following way:

$$\rho_j^1 = \rho_j^0 + \nu\left(\rho_{j+1}^0 - 2\rho_j^0 + \rho_{j-1}^0\right) \qquad (2.3.8)$$

For the zero-velocity case the diffusion coefficient is strictly 0.125. In the case of nonzero velocities ν is 0.125 plus small wave-number and wave-number dependent terms. A von Neumann analysis of this equation shows that the amplification factor is always less than unity for $\Sigma <$ 0.5, so the transport scheme is always stable.

Stage II: Antidiffusive transport

It is straightforward to remove the zero-order diffusion from stage I by applying an equal and opposite anti-diffusion. Since the previous equation represents an implicit tridiagonal system for $\{\rho_j^0\}$ it may readily be inverted by the use of standard algorithms. However, the fastest and simplest way to remove this residual diffusion is by an explicit anti-diffusion equation

$$\bar{\rho}_j^1 = \rho_j^1 - \tfrac{1}{8}\left(\rho_{j+1}^1 - 2\rho_j^1 + \rho_{j-1}^1\right) \qquad (2.3.9)$$

and this is the scheme we implement.

This equation has the disadvantage of not conserving positivity. Consider the example shown in Figure 2.2 . The antidiffusion imposed in stage II, which is only intended to remove numerical errors introduced in stage I, in fact brings in new numerical errors at the labelled grid points. New maxima and minima are formed where they are physically unreasonable, and the new minimum is negative. We cure these features by invoking the principle of *flux corrected transport*.

Stage III: Flux correction

To remove the nonpositivity it is more direct to work with the mass fluxes directly. We may rewrite the antidiffusion formula

$$\bar{\rho}_j^1 = \rho_j^1 - f_{j+\frac{1}{2}} + f_{j-\frac{1}{2}} \qquad (2.3.10)$$

where the antidiffusive mass fluxes are defined as

$$f_{j\pm\frac{1}{2}} \equiv \pm \frac{1}{8}\left(\rho_{j\pm1}^1 - \rho_j^1\right) \qquad (2.3.11)$$

This rewritten antidiffusion formula has two important properties:

i) The antidiffusive fluxes $f_{j+\frac{1}{2}}$ and $f_{j-\frac{1}{2}}$ describe explicit transfers of material.

ii) The equation is strictly conservative, no matter what values the fluxes take, as every flux is added once and subtracted once somewhere else (except at the boundaries).

We are thus led to the principle of flux-corrected transport:

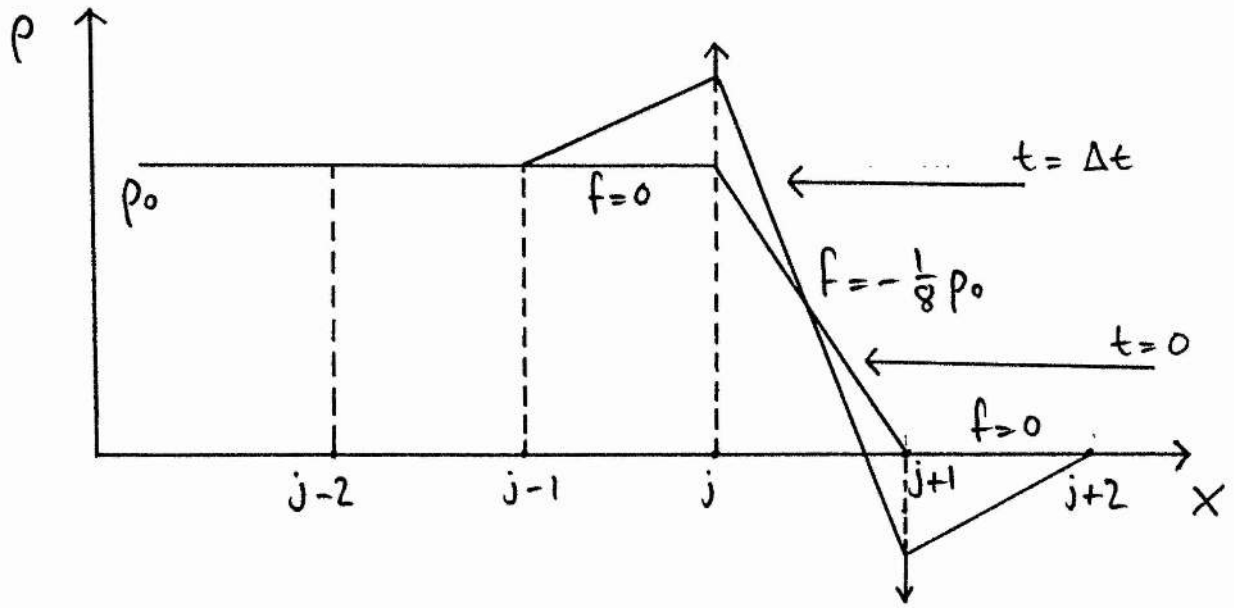


Figure 2.2 Effects of uncorrected antidiffusion

The antidiffusion stage should generate no new maxima or minima in the solution, nor should it accentuate already existing extrema.

This scheme clearly maintains positivity. To make the limitation quantitative without violating conservation, we correct the antidiffusive mass fluxes. The $\{f_{j+\frac{1}{2}}\}$ are limited term by term so that no antidiffusive-flux transfer of mass can push the density value at any grid point beyond the density value at neighbouring points. This is the origin of the name 'flux-corrected transport' and is the crux of the method.

The corrected fluxes $\{f^c_{j+\frac{1}{2}}\}$ are given by the formula

$$f^c_{j+\frac{1}{2}} = \mathrm{sgn}\,\Delta_{j+\frac{1}{2}} \max\{0, \min(\Delta_{j-\frac{1}{2}}\,\mathrm{sgn}\,\Delta_{j+\frac{1}{2}}, \text{“}\tfrac{1}{8}\text{”}\,|\Delta_{j+\frac{1}{2}}|, \Delta_{j+\frac{3}{2}}\,\mathrm{sgn}\,\Delta_{j+\frac{1}{2}})\} \qquad (2.3.12)$$

where

$$\Delta_{j+\frac{1}{2}} \equiv \rho^1_{j+1} - \rho^1_j \qquad (2.3.13)$$

and the fluxes $\{f_{j+\frac{1}{2}}\}$ are replaced in the original antidiffusion equation. The quotation marks about the factor 0.125 indicate that a more exact cancellation of factors can be achieved if even rough approximations to the velocity and wavenumber dependent corrections are included.

An application of this schema to sharp gradients gives rise to the phenomenon of 'flux clipping' or flux limiting. Even if a shock wave exhibits this behaviour, the accuracy of the SHASTA algorithm is such

that the Rankine-Hugoniot relations (Jeffrey and Taniuti, 1964) are still satisfied.

2.4 Solution of the diffusion equation

There exist several ways to solve the two-dimensional diffusion equation

$$\frac{\partial \underline{B}}{\partial t} = \eta \nabla^2 \underline{B} \qquad (2.4.1)$$

of which the best known is the Alternating-Direction Implicit (ADI) (Mitchell, 1969). The scheme we choose here is also implicit and has the advantage, as do all this class of algorithms, of being unconditionally stable so that a timestep of any size may be taken without loss of numerical stability. Differencing the x-component of (2.4.1) gives

$$\frac{B_j^{n+1} - B_j^n}{\Delta t} = \eta \frac{B_{j+1}^{n+1} - 2B_j^{n+1} + B_{j-1}^{n+1}}{\Delta x^2} \qquad (2.4.2)$$

This expression is first order in time and second order in space. Rewriting (2.4.2),

$$B_{j-1}^{n+1} - \left(2 + \frac{\Delta x^2}{\eta \Delta t}\right) B_j^{n+1} + B_{j+1}^{n+1} = -\frac{\Delta x^2}{\eta \Delta t} B_j^n \qquad (2.4.3)$$

showing that $\{B_j^{n+1}\}$ may be found by solving the matrix equation (ignoring the boundary conditions)

$$\begin{bmatrix} -\left(2+\frac{\Delta x^2}{\mu \Delta t}\right) & 1 & 0 & \cdots \\ 1 & -\left(2+\frac{\Delta x^2}{\mu \Delta t}\right) & 1 & \cdots \\ 0 & \cdots & & \\ & & & \\ & \cdots & 0 & 1 & -\left(2+\frac{\Delta x^2}{\mu \Delta t}\right) \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}^{n+1} = -\frac{\Delta x^2}{\mu \Delta t} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}^{n} \qquad (2.4.4)$$

This may be easily solved on noting that the matrix is tridiagonal. There exist several explicit, non-iterative algorithms to invert such matrices; we choose that of Crout (Burden *et al*, 1981).

For a numerical scheme of this type the boundary conditions must be imposed implicitly on the form of the matrix, unlike the advective part of the code where they must be imposed explicitly. For instance, consider the case of a zero first-order derivative boundary condition. Denote the entry in the ith row and jth column by a(i,j); then the matrix may be represented in the form [a(i,j)]. To impose a zero normal derivative at the upper boundary, let $B_1^{n+1} = B_3^{n+1}$, $B_2^{n+1} = 0$. The matrix and vectors of magnetic field will then assume the form

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots \\ -a_{23} & a_{22} & a_{23} & 0 \cdots \\ 0 & a_{32} & a_{33} & \cdots \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} B_1 \\ 0 \\ B_3 \\ B_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} B_1 \\ 0 \\ B_3 \\ B_4 \\ \vdots \end{bmatrix} \qquad (2.4.5)$$

where the vectors are the values of magnetic field on a column of the numerical grid. The reader will note that expanding the second row gives

$$- a_{23} B_1^{n+1} + a_{23} B_3^{n+1} = 0$$

$$\Rightarrow B_1^{n+1} = B_3^{n+1}$$

as required.

2.5 Spatial differencing

The computation is performed in Cartesian coordinates over a rectangular array of 40×60 grid points. The points are uniformly spaced in the x-direction but are non-uniformly spaced in the z-direction to ensure higher resolution at one edge of the box. The i'th grid point in the z-direction is located at

$$z_i = \left[(1+f)^{i-2} - 1\right] / \left[(1+f)^{nz-2} - 1\right] \qquad (2.5.1)$$

where nz is the number of points along the z-axis (= 60) and f is the fractional increase in spacing from one grid-point to the next (= 0.10). The lowest grid spacing is thus $\Delta z \approx 0.012$ and the highest is $\Delta z \approx 0.133$. We found from experiment that f could not exceed about 0.13 or the code went unstable.

We are limited by the grid resolution as to the maximum magnetic Reynolds number that may be taken. In the *grid separation instability*, a cascade of energy proceeds from long wavelengths to short ones due to the nonlinearity of the MHD equations; the physics '*literally slips*

through the cracks' (Pearson *et al*, 1985). This need not affect the behaviour of the code as long as we can be sure that all wavelengths are being modelled accurately.

One way to do this is to use a spectral code, in which the MHD equations are Fourier analysed in space and the governing equations in Fourier space are solved numerically. Such a scheme ensures that all length scales can be accurately described throughout the calculation as long as computation is stopped when significant activity appears in the highest wavenumbers. The difficulty with such a scheme is that the treatment of boundary conditions becomes complicated for any but the simplest configurations, with one analytical expression required for each harmonic in both directions. Since hundreds of harmonics are required for problems of interest, this approach presents some rather severe difficulties. Matthaeus and Montgomery (1981) have performed a study of the double tearing mode in two dimensions with such a spectral code, the advantage being that for such a configuration only symmetry conditions need be imposed on the boundary. They find interesting evidence of current filamentation, a feature observed in some laboratory experiments (Stenzel *et al*, 1986) but not in any finite-difference calculations performed to date. A spectral approach is also suitable for a study of homogeneous turbulence.

We choose instead to follow the approach of Forbes and Priest (1983), who attempt to model all length scales within a finite-difference scheme and thus to avoid grid-separation instability. They argue that the numerical grid spacing should be less than the Kolmogorov dissipation scale lengths λ

$$\lambda_\eta = \left| \frac{d\Sigma_B}{dt} \eta^{-3} \right|^{-\frac{1}{4}} \qquad \lambda_\nu = \left| \frac{d\Sigma_v}{dt} \nu^{-3} \right|^{-\frac{1}{4}}$$

(Landau and Lifschitz, 1959, p 122) where $\mathcal{E}_B$ and $\mathcal{E}_V$ are the total magnetic and kinetic energies, respectively. In our equations the viscosity arises only from numerical effects and the magnetic conductivity is the major significant source of diffusion. We therefore ensure that the grid-resolution is sufficient at the current sheet to cover the shortest wavelengths arising at any time.

Eastwood and Arter (1985) have performed a important and controversial study of this type of instability in the context of simulations of tokamak disruptions. They perform a detailed analytical study of aliasing error,a common cause of 'blow-up' of solutions. As an illustration they consider the discretised, Fourier transformed fluid equations. Let the shortest wavelength have wavenumber K and let the velocity spectrum be filled for k up to K; then the non-linear advection terms attempt to generate flows with wavenumbers up to 2K. Those in the range K + 1 to 2K pretend to be, or 'alias', flows of wavenumber less than K. If the physical process being modelled is a cascade of energy from large to small scales, a positive feedback instability is possible, since energy which should have been transferred to small length scales is instead transferred to long ones. The instability usually manifests itself first as a violation of energy conservation followed by the appearance of arbitrarily large velocities and magnetic fields so that the timestep (here determined by the CFL condition) drops towards zero. However, identical behaviour is seen when there is a mathematical or programming error in the boundary conditions or vector potential conversion routines, so great care must be taken in deciding which of these is causing a simulation to fail. Roache (1976) warns specifically

against mistaking non-linear instabilities for effects caused by programming errors in the context of fluid-dynamical simulations.

It seems likely that this instability caused Forbes and Priest's study of line-tied reconnection to end prematurely. We have also observed the effect in some of our tearing mode simulations, which are very similar to theirs. Chapter 4 describes how one physically allowable perturbation, imposed to initiate magnetic tearing, drove the code unstable in a few Alfven times while others let the code run indefinitely. In the light of these ideas we suggest that the first perturbation contained harmonics at higher wave-numbers, so that the cascade process for the first run took much less time to reach the aliasing stage than the second.

Boundary conditions are imposed explicitly by the modules BOUNDY and implicitly from within AFROMB, BFROMA and DIFUSE. These are listed in Appendix B.

2.6 Temporal differencing

The timestep Δt is determined by the Courant-Friedrichs-Lewy condition (Potter, 1981)

$$\Delta t = A\,\Delta / \left[\left(V_a^2 + C_s^2\right)^{\frac{1}{2}} + \left(V_x^2 + V_z^2\right)^{\frac{1}{2}} \right] \qquad (2.6.1)$$

where Δ is the smallest grid spacing in the mesh (= min(Δx, Δz)), Vx and Vz are the x and z components of the plasma velocity, and is the fast mode wave speed. A is the Courant number which is set to 0.2 to ensure second-order accuracy and linear numerical stability. The timestep is determined at every iteration in the subroutine VTMSTP.

CHAPTER 3

DRIVEN MAGNETIC RECONNECTION

3.1 Introduction

In this chapter we apply the computer program described in Chapter 2 to a study of steady-state driven magnetic reconnection. Because of the non-linearity of the MHD equations and the resulting difficulty in finding analytical solutions under general boundary conditions, most work performed to date has been numerical (Birn, 1985; Biskamp, 1986; Forbes and Priest, 1982, 1983, 1984; Lee and Fu, 1986; Matthaeus and Montgomery, 1981; Sato and Hayashi, 1979; Ugai, 1984; Ugai and Tsuda, 1977). The results, reviewed in Priest (1985) have shown many new and unexpected features, such as the rapid creation and annihilation of neutral points found by Forbes and Priest (1982 , 1983a, b) and the emergence of plasma jets from a Y-point at the end of a current sheet in Biskamp's simulation (1986), as described and analysed by Soward and Priest (1986). However, the effect of varying the boundary conditions and internal paramaters remains poorly understood.

Chapter 3 presents the background and results to a series of numerical experiments in which three externally adjustable parameters are systematically varied and their effect on the solutions examined. In section 3.2 we describe the numerical implementation of the driven reconnection problem together with an account of the development of the code. In section 3.3 a 'standard run' with typical magnetic Reynolds number and boundary conditions is analysed in detail. In sections 3.4, 3.5 and 3.6 we show the effect of varying incoming field line curvature,

plasma resistivity and plasma inflow speed. Sections 3.7 and 3.8 have some general comments on the dynamics of driven reconnection arising from this study.

3.2 Numerical implementation

The field topology we choose to study has two symmetrical, adjacent, two-dimensional antiparallel flux systems with externally imposed flow and field curvature. Such a configuration might occur inside a tokamak when two islands with different m, n numbers overlap and coalesce, or in the Earth's geomagnetic tail during a substorm (Figures 3.1, 3.2). We chose to impose symmetry in both the x- and y-directions, which may be exploited to reduce the number of grid points required, and hence the computer time needed for each run. Instead of considering the whole system (Figure 3.3), we may instead examine only one quadrant (Figure 3.4). This precludes the occurrence of some asymmetric tearing phenomena but will otherwise give exactly the same information.

For convenience, we assign numbers to label each of the four numerical boundaries, as shown on figure 3.4. In analytical form the boundary conditions are

Surface 1 (symmetry):

$$B_z = v_x = \frac{\partial B_x}{\partial x} = \frac{\partial v_z}{\partial x} = \frac{\partial \rho}{\partial x} = \frac{\partial \rho}{\partial x} = 0 \qquad (3.2.1)$$

Surface 2 (symmetry):

$$B_x = v_z = \frac{\partial B_z}{\partial z} = \frac{\partial v_x}{\partial z} = \frac{\partial \rho}{\partial z} = \frac{\partial \rho}{\partial z} = 0 \qquad (3.2.2)$$

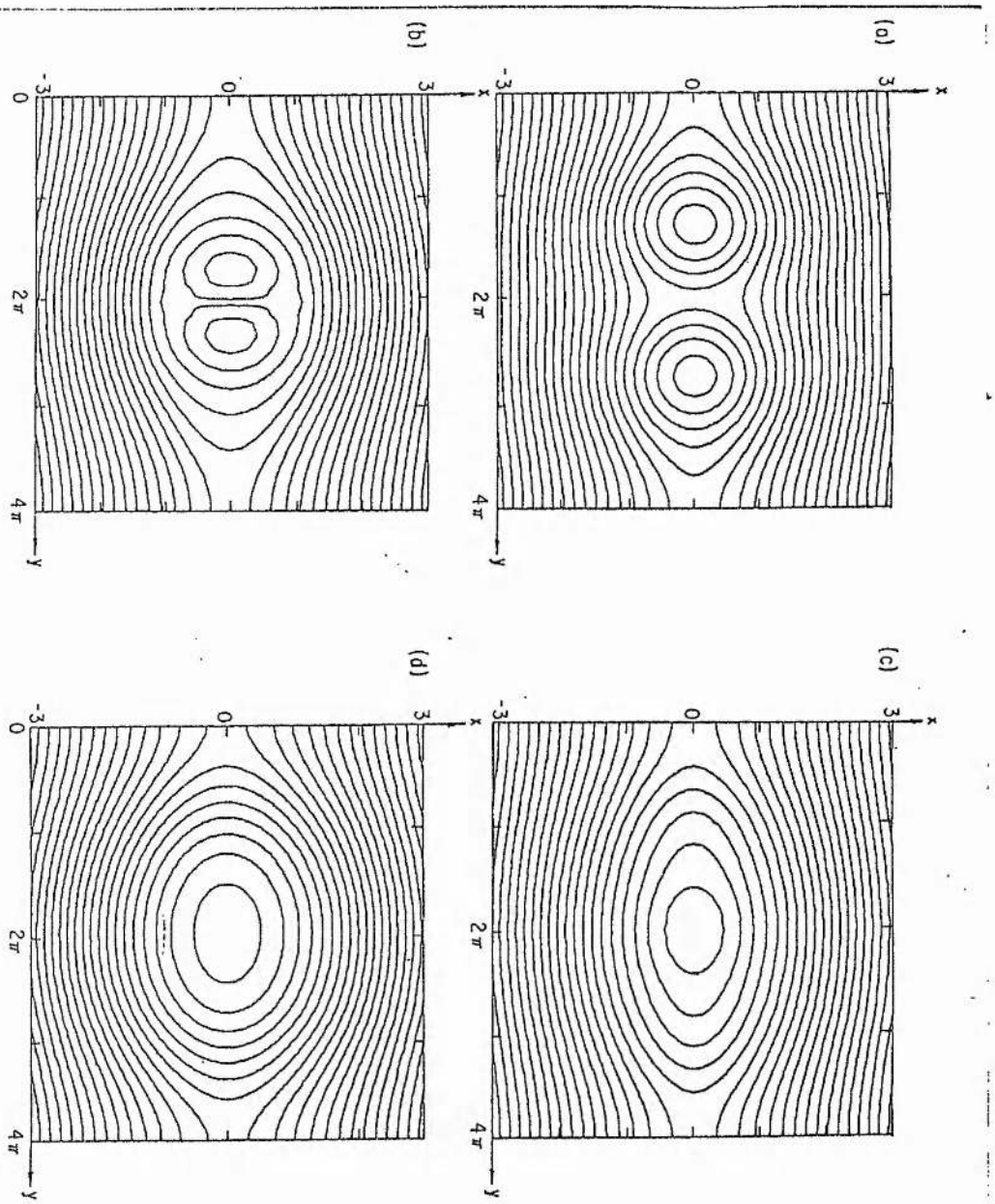


Figure 3.I Coalescence of magnetic islands

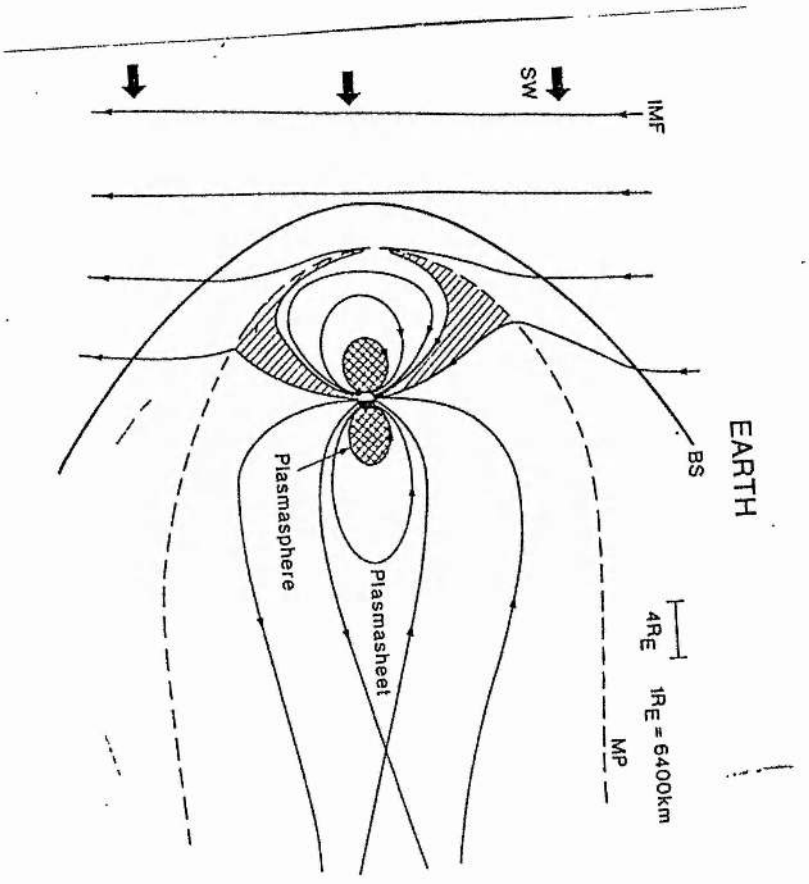


Figure 3.2 Reconnection in the Earth's magnetosphere

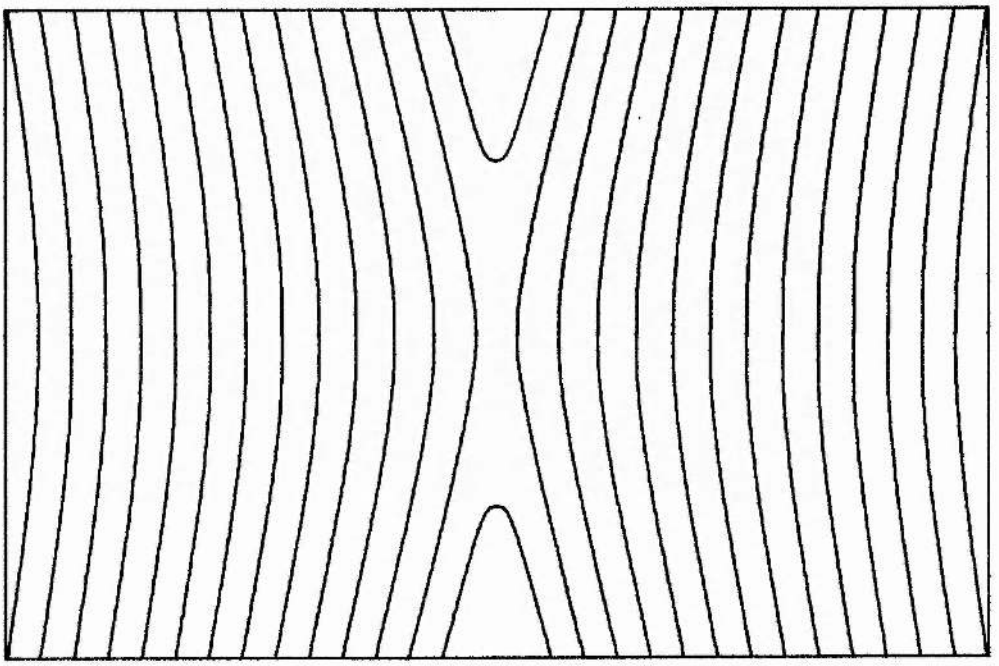


Figure 3.3 Total magnetic field configuration

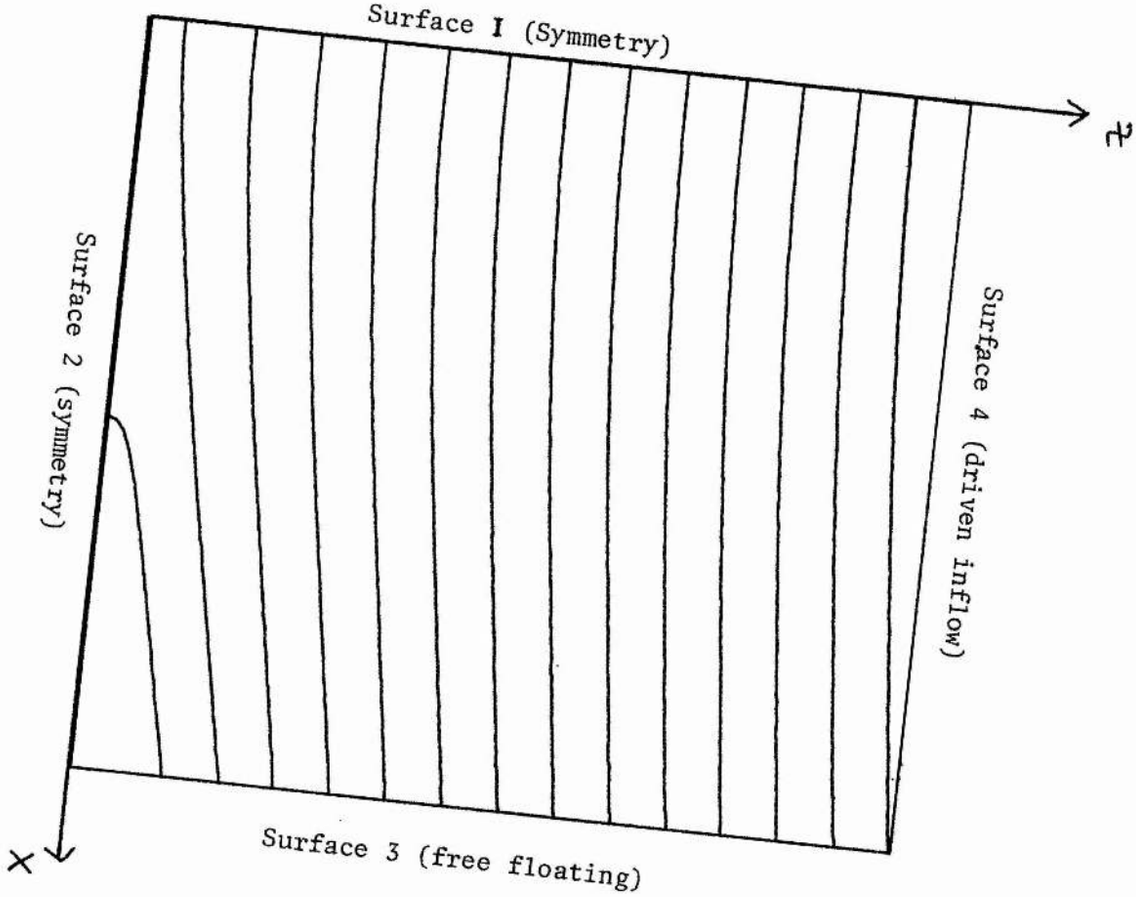


Figure 3.4 Single quadrant of system

Surface 3 (freefloating):

$$\frac{\partial B_z}{\partial x} = \frac{\partial v_x}{\partial x} = \frac{\partial v_z}{\partial x} = \frac{\partial p}{\partial x} = \frac{\partial \rho}{\partial x} = 0 \qquad \frac{\partial B_x}{\partial x} = -\frac{\partial B_z}{\partial z} \tag{3.2.3}$$

Surface 4 (driven inflow):

$$B_x,\ B_z,\ v_x,\ v_z,\ p,\ \rho \qquad \text{imposed} \tag{3.2.4}$$

where the variables and their non-dimensionalisation are defined in Chapter 2.

To use an analytical expression for a boundary condition, we must find a numerically equivalent condition so that the code will assign meaningful values to the gridpoints at the edge of the numerical box. For an axis of symmetry this is straightforward; we define the row or column of points one gridpoint in from the boundary to be the edge of the system, and calculate the values of all independent variables at the edge on the basis of their values two grid-points in. Specifying the subscript on a variable as the index along the normal to each surface, the boundary conditions for edges 1 and 2 are implemented as

Surface 1 (symmetry)

$$\begin{aligned} B_{x_1} &= B_{x_3} & v_{x_1} &= -v_{x_3} & p_1 &= p_3 \\ B_{z_1} &= -B_{z_3} & v_{x_2} &= 0 & \rho_1 &= \rho_3 \\ B_{z_2} &= 0 & v_{z_1} &= v_{z_3} & & \end{aligned} \tag{3.2.5}$$

Surface 2 (symmetry)

$$\begin{aligned} B_{x_1} &= -B_{x_3} & v_{x_1} &= v_{x_3} & & \\ B_{x_2} &= 0 & v_{z_1} &= -v_{z_3} & p_1 &= p_3 \\ B_{z_1} &= B_{z_3} & v_{z_2} &= 0 & \rho_1 &= \rho_3 \end{aligned} \tag{3.2.6}$$

To impose the values of all variables at the inflow boundary is similarly straightforward (although the reader should note the risk of overspecification mentioned in Section 2.3.3 and in Appendix A).

Surface 4 (driven inflow)

$$B_{x_1} = B_{x_{in}} \quad V_{x_1} = V_{x_{in}} \quad \rho_1 = \rho_{in}$$

$$B_{z_1} = B_{z_{in}} \quad V_{z_1} = V_{z_{in}} \quad \rho_1 = \rho_{in} \qquad (3.2.7)$$

The implementation of a free-floating boundary illustrates a limitation of the finite-difference technique. For an analytical analysis of an astrophysical plasma a boundary may safely be taken to be at infinity, but the computational domain must of necessity have a finite number of gridpoints. One way to overcome this is to isomorphically map the finite computational domain onto an infinite region, as did Weber (1978). However, the resolution becomes correspondingly poor towards infinity so that an MHD wave propagating into this region will eventually trigger a numerical instability (Forbes, 1987). Alternatively, a Sommerfeld radiation condition

$$\frac{\partial \phi}{\partial t} + V_p \frac{\partial \phi}{\partial x_n} \qquad (3.2.8)$$

may be used, where ϕ is a flow variable and V_p is the magnitude of the local phase speed of the wave quantity ϕ (Orlanski, 1976; Han et al, 1983). Forbes (1987) observes that this condition forces the wave at the boundary to be outward propagating, giving a well-posed initial-boundary-value problem. Despite this, numerical methods based on the full Sommerfeld condition may be prone to instability (Israeli and Orszag, 1981).

We choose instead to implement the much simpler boundary condition of first-order extrapolation of all variables:

$$\frac{\partial \phi}{\partial x_n} = 0 \qquad (3.2.9)$$

where x_n is the coordinate perpendicular to the surface under consideration. Chu and Sereney (1974) have performed a comparison of seven different methods of formulating boundary conditions at an outflow boundary for a one-dimensional hydrodynamical open-boundary problem. They found that only first-order extrapolation and a method based on extrapolation of the characteristics (equivalent to the Sommerfeld condition) gave results consistent with the known analytical solution. The other, more complex methods gave less accurate results. Hence we believe that this method is one of the best for the problem.

Given time, we would have improved these boundary conditions by using the very similar method described by Camerlengo and O'Brien(1980):

$$\phi_k^{n+1} = \phi_k^n \quad \forall p \leq 0$$
$$\phi_k^{n+1} = \phi_{k-1}^n \quad \forall p > 0 \qquad (3.2.10)$$

If the local wave is incoming, ϕ does not change over an iteration, while if the local wave is outgoing ϕ is given by first order extrapolation in space and time, approximating a Sommerfeld condition. Alternatively a simple extrapolation may be combined with a wave-absorbing region at the boundary (Bayliss and Turkel, 1982) which upholds the 'no wave-reflection' property of the Sommerfeld condition.

The disadvantage is that a large number (at least 10%) of the gridpoints must lie in this region, which is not modelling any physical processes.

The initial conditions are

$$V_x = 0$$

$$V_z = 0$$

$$B_x = \sin (\pi z / 2 w) \quad (z < w)$$

$$B_x = 1 \quad (z > w) \qquad (3.2.11)$$

$$B_z = 0$$

$$\rho = 1$$

$$p = (1 + \beta - |\underline{B}||\underline{B}|)/2$$

where w is the width of the initial current sheet (= 0.05) and β, the ratio of gas to magnetic pressure, is 0.1. The pressure condition ensures the system is in magnetohydrostatic equilibrium at t = 0.

3.2.1 Vector potential conversion routines

We choose to use the magnetic vector potential A as a supplementary variable. Several advantages arose from this. Only A need be treated in the the diffusive part of the program, rather than both components of magnetic field, with a resulting saving in computing costs. Since $\underline{B}$ = curl A, elementary vector identities ensure that div $\underline{B}$ = 0 identically at all times. However, routines to convert from A to $\underline{B}$ and back at every iteration were still required since the advective equations are cast in terms of $\underline{B}$. Also, more complex boundary conditions that impose A, B_x and B_z selfconsistently were required, and finding

workable expressions (especially at the corners of the box) proved to be extremely difficult.

The calculation of $\underline{B}$ from A is straightforward. A two point formula is used to obtain the derivative of A in the x direction:

$$\frac{\partial A}{\partial x} = \frac{A_{i+1} - A_{i-1}}{x_{i+1} - x_{i-1}} \qquad (3.2.12)$$

This gives full second order accuracy on the uniformly spaced axis. For a similar accuracy order to hold in the z-direction a four point formula would be required (Roache, 1976, p 20). We decided that the associated complexity of the associated boundary conditions would have been too expensive in development costs and implemented another two point formula (Schnack, 1978):

$$\frac{\partial A}{\partial z} = 2\,\frac{A_{j+1} - A_{j-1}}{\Delta_+ z - \Delta_- z} \qquad (3.2.13)$$

where

$$\Delta_+ z = z_{j+1} - z_j$$
$$\Delta_- z = z_j - z_{j-1} \qquad (3.2.14)$$

The truncation error is

$$e_T = -\frac{1}{4}A''\,\frac{\Delta_+ z^2 - \Delta_- z^2}{\Delta_+ z + \Delta_- z} - \frac{1}{24}A'''\,\frac{\Delta_+ z^3 + \Delta_- z^3}{\Delta_+ z + \Delta_- z} \qquad (3.2.15)$$

so that (3.2.13) is formally first order accurate, becoming second order only on a uniform grid. However, our grid spacing varies as

$$\frac{\Delta_+ z}{\Delta_- z} = r \qquad (3.2.16)$$

where $r = 1 + \varepsilon$, ε a small constant. In this case (3.2.15) becomes

$$e_T = -\frac{1}{4} A'' \frac{r^2 - 1}{r+1} \Delta_- z - \frac{1}{24} A''' \frac{r^3 + 1}{r+1} \Delta_- z^2 \qquad (3.2.17)$$

The coefficient of the first order term behaves like

$$\frac{2\varepsilon + \varepsilon^2}{2 + \varepsilon} = \varepsilon << 1 \qquad (3.2.18)$$

and thus we have 'effective' second order accuracy.

To derive A from $\underline{B}$ proved to be a considerably more complex problem. To calculate A one of the two first-order differential equations

$$B_x = -\frac{\partial A}{\partial z} \qquad B_z = \frac{\partial A}{\partial x}$$

must be integrated subject to the boundary conditions, and the first difficulty becomes apparent: if the vector potential is integrated up a column or along a row two boundary conditions for A have to be imposed on the solution of a first order system, leading to overspecification.

The first approach tried for the driven reconnection problem (and the one eventually adopted) was straightforward integration of the x-component of magnetic field in the z-direction:

$$A(x,z) = \int_0^z B_x(x,z')\, dz' \qquad (3.2.19)$$

The value of the time-dependent vector potential was specified at the inflow boundary and at the two rows further in, and the field was integrated down to the bottom of the box. To maintain second order accuracy $\underline{B}$ was integrated over alternate pairs of points, an approach used by Forbes and Priest (1982) . To check for consistency, the two conversion routines BFROMA and AFROMB were run alternately on the intial magnetic field profile, and to within single precision VAX arithmetic the results were consistent to 6 decimal places after several hundred iterations.

With low values of R_m this approach worked well, but decreasing η gave some noise in the field lines, which are here plotted as contours of constant A. The difficulty is probably purely numerical, in that the equations which convert $\underline{B}$ from A are not being inverted in precisely the same form. That is, while a three point formula is used to differentiate A, a two point formula derives A from $\underline{B}$ and so 'decoupling' can occur between the values of A at odd and even rows. This gives rise to the noise, which is caused by rapid alternations of A from gridpoint to gridpoint. This nonphysical effect is probably not seen when low values of Rm are taken because the associated high Fourier modes and steep gradients are damped out by the high resistivity. It is possible that this is the cause of some of the numerical instabilities seen by Forbes in his work (1982, 1983a, b), as well as the Kolmogorov length-scale and aliasing instabilities discussed in Chapter 2, section 5.

An obvious further approach would be to modify the vector potential routines so that the differential equations are integrated exactly. This would require a tridiagonal matrix to be inverted for every column.

Although cheap to run, we did not have enough time to write and debug this extension to the code.

3.2.2 Compatibility relations

The MHD equations we solve in this chapter (2.1.1 - 2.1.4) are of mixed hyperbolic-parabolic character, since the introduction of dissipation adds second-order terms to the system. With even a partly hyperbolic system, the need to assign values to each of the variables on boundaries over which there is a mass and field flux will lead to over-specification and the consequent formation of a boundary layer. This section describes a consistent formulation to avoid over-specification by determining the values of some of the boundary variables from the system of equations being solved. The method proceeds through determining the characteristic equations of the system and transforming them to lie along the projected characteristics as described in Appendix A. A subset of the resulting compatability relations may then be solved; the number will depend on how many variables cannot be specified at that point, which in turn equals the number of characteristics leaving the computational domain (Courant and Hilbert, 1962, p 473).

Hu and Wu (1985) give a full discussion of the characteristics of the two-dimensional ideal (purely hyperbolic) equations. Here we may consider the simpler case of a one-dimensional scan of the ideal two-dimensional equations; ideal, because we assume resistive effects to be unimportant everywhere away from the x-point; one-dimensional, because our numerical scheme scans in the x- or z-direction while ignoring derivatives in the perpendicular coordinate. Since our system is in

Cartesian geometry the projected characteristics will just lie along the n-t boundary, where n is the unit normal to the boundary at the specified point. We restrict further analysis to the z-t plane since the characteristic equations are primarily of interest to us in the context of the driven inflow boundary (z = 0), and refer the reader to Appendix A for a full treatment of the problem. For reference, we observe here that for the inflow regime under consideration ($-V_s \leq V_{in} \leq 0$) only one compatability relation need be solved, which is

$$V_s B_x B_z D_t V_x - V_s \rho (V_s^2 - V_a^2) D_t V_z + \alpha^2 B_x D_t B_x - V_s^2 B_x V_x \frac{\partial B_x}{\partial z} + (V_s^2 - V_a^2) D_t \rho = 0 \qquad (3.2.20)$$

where

$$D_t = \frac{\partial}{\partial t} + (V_z - V_s) \frac{\partial}{\partial z} \qquad (3.2.21)$$

3.2.3 Program development

Given the problem as defined in the previous sections, a suitable combination of initial and boundary conditions had to be found which allow the system to evolve in time to give asymptotically steady-state reconnection. For development purposes we omitted the use of any compatability relations, as they invariably complicated the behaviour of the system to such an extent that it was unclear which effects were physical and which numerical. For instance, a crude approximation to the compatability relation was obtained by setting

$$\frac{\partial \rho}{\partial x_n} = 0 \qquad (3.2.22)$$

where X_n is the coordinate normal to the inflow boundary. This boundary condition led to a rapid drop in plasma pressure across the inflow boundary and instability in the numerical scheme.

The first set of initial and boundary conditions tried were a plane-parallel current sheet with no x-dependence in any of the variables, and a similar set of inflow conditions:

$$V_x = 0$$

$$V_z = V_{in};\ V_{in} = 0.05$$

$$B_x = 1$$

$$B_z = 0$$

$$\rho = 1$$

$$p = \beta/2;\ \beta = 0.1$$

Ugai and Tsuda have performed a numerical experiment (1979a, b) with the same initial conditions. The experiment was identical in nearly all respects to ours, save for the crucial differences that inflow speed could not be specified and a region of anomalously low magnetic conductivity was assumed about the origin. This configuration set up a plasma jet and a strong B_z component along the x-axis, indicating that fast reconnection had been initiated. On the basis of their experiment we assumed that ours would show the same behaviour, but this was not the case; the system developed purely one-dimensionally. While initial, transient discontinuities propagated in the z-direction, no x-dependent behaviour was seen.

Without the region of anomalous conductivity, this behaviour may be explained on physical and numerical grounds. Clearly, forcing a fluid

over one boundary into a box bordered by two rigid boundaries and another open boundary will lead to an outflow over this last boundary. Unfortunately, the lack of asymmetry assumed in the initial conditions is propagated in time due to the form of the numerical scheme; what we are in fact solving is a row of identical one-dimensional MHD systems. Such a simulation corresponds to pushing together two infinite antiparallel flux systems, for which no z-component of the magnetic field would appear.

It was thus clear that some degree of asymmetry had to be introduced even though the simulation would not then be strictly comparable with most analytical studies of reconnection. We tried: (1) the above set of inflow conditions with an imposed z-component of the field in the initial state (initially curved field lines); (2) the above set of inflow conditions with an imposed x-component of the plasma velocity in the initial state (initially imposed outflow velocity); (3) the above initial conditions with an x-dependent V_z (asymmetric inflow velocity profile). After the expected initial transients, all these cases settled down to essentially one-dimensional behaviour (no z-component of the magnetic field appeared about the x-axis) and it became clear that the imposed asymmetry had to be in the z-component of the magnetic field. That is, the incoming field lines had to be curved. This is common to all other driven reconnection simulations we have seen (Biskamp, 1986; Sato, 1979), although the authors do not state the reason. We interpret this as a consequence of the low plasma beta, for which the magnetic field dominates the flow. Increasing β by one or more more orders of magnitude (making the system more hydrodynamic) gave acceptable flow patterns but led to little interesting MHD behaviour.

The next step was therefore to try the inflow conditions

$$V_x = 0$$

$$V_z = V_{IN};\ V_{IN} = 0.05$$

$$B_x = 1$$

$$B_z = \alpha \sin (\pi x /2)$$

$$\rho = 1$$

$$p = \beta/2;\ \beta = 0.1$$

where the real parameter α was adjustable, and with a plane parallel initial configuration. This did give an x-point in the flow with the expected two-dimensional flows. However, if α exceeded about 0.1 the code went unstable, due to a sharp gradient in the z-component of the magnetic field.

We then found that while the simulation ran as expected for the first few Alfven transit times, the transient shock waves generated by imposing the inflow conditions at t = 0 did not damp out but grew in amplitude, eventually obscuring all features of interest. We cured this by introducing a further asymmetry into the inflow, making V_z a function of x, which involved a readjustment of B_x in such a way that the electric field at the boundary ($E = V_z \times B_x$) remained uniform.

We next introduced a time dependence into the inflow conditions. The inflow velocity, V_z, was linearly ramped over two Alfven times from zero to its full value to diminish further the effect of the initial transient shock waves, a method which has been widely used elsewhere (Biskamp, 1986; Sato, 1979). The inflow value of B_z was also ramped in time, as we found this to be the only way to reach sufficient values of

this variable for the slow mode shock waves at the current sheet to be clearly demarcated. The extra advantage was gained of allowing α to reach values of 0.3 without instability.

The inflow conditions finally adopted are

$$V_x = 0$$

$$V_z = V_{in}\, r(t) \cos (\pi x / 10)$$

$$B_x = 1 / \cos (\pi x / 10)$$

$$B_z = \alpha\, r(t) \sin (\pi x / 2)$$

$$\rho = 1$$

$$p = \beta/2$$

$$r(t) = t / 2 \qquad (t < 2)$$

$$r(t) = 1 \qquad (t > 2)$$

The reader will notice that no compatability relations are being used. When the program was being developed a shock wave formed over the first two or three grid points in from the boundary. This was as expected, as at this stage the inflow variables were overspecified and a boundary layer should have appeared, analogous to the formation of a boundary layer along a solid-liquid interface in fluid mechanics. When a simplified compatability relation was included for the pressure, so that p was allowed to 'free float' at the inflow boundary, the value of p quickly dropped to zero. The resulting behaviour was clearly nonphysical. A magnetic pressure gradient formed opposing the inward motion of the plasma, and the z-velocities at one grid point in from the boundary were seen to be in the opposite direction from those imposed by the program at the boundary.

The reasons for this are still not clear. It is possible that the compatability relations were oversimplified to such an extent that the desired behaviour was not seen. If much more time had been available, it would certainly have been desirable to implement a full compatability relation.

3.3 A typical run

In the remaining part of this chapter we examine the dynamics of an externally driven reconnecting system by comparing some numerical solutions of the asymptotically steady-state MHD equations. We start with a close examination of one 'standard' run (Rm = 1000, α = 0.1, V_{in} = 0.05). For completeness a full set of plots of all steady-state field variables is given in diagrams (3.5) - (3.15), comprising:

(3.5) Contours of magnetic vector potential

(3.6) Plasma flow vectors

(3.7) Magnetic field (B_x), pressure (p), velocity (V_z), and density (ρ) on the z-axis

(3.8) B_x, p, V_z, ρ on the line x = 0.5

(3.9) B_z, p, V_x, ρ on the x-axis

(3.10) B_z, p, V_x, ρ on the line z = 0.5

(3.11) Current density J (= curl $\underline{B}$) plotted against x and z

(3.12) Current density J plotted against x and z; opposite orientation to (3.11)

(3.13) Plasma pressure plotted against x and z

(3.14) Electric field at the neutral point (x, z = 0) as a function of time

(3.15) Electric field at the end of the calculation plotted against

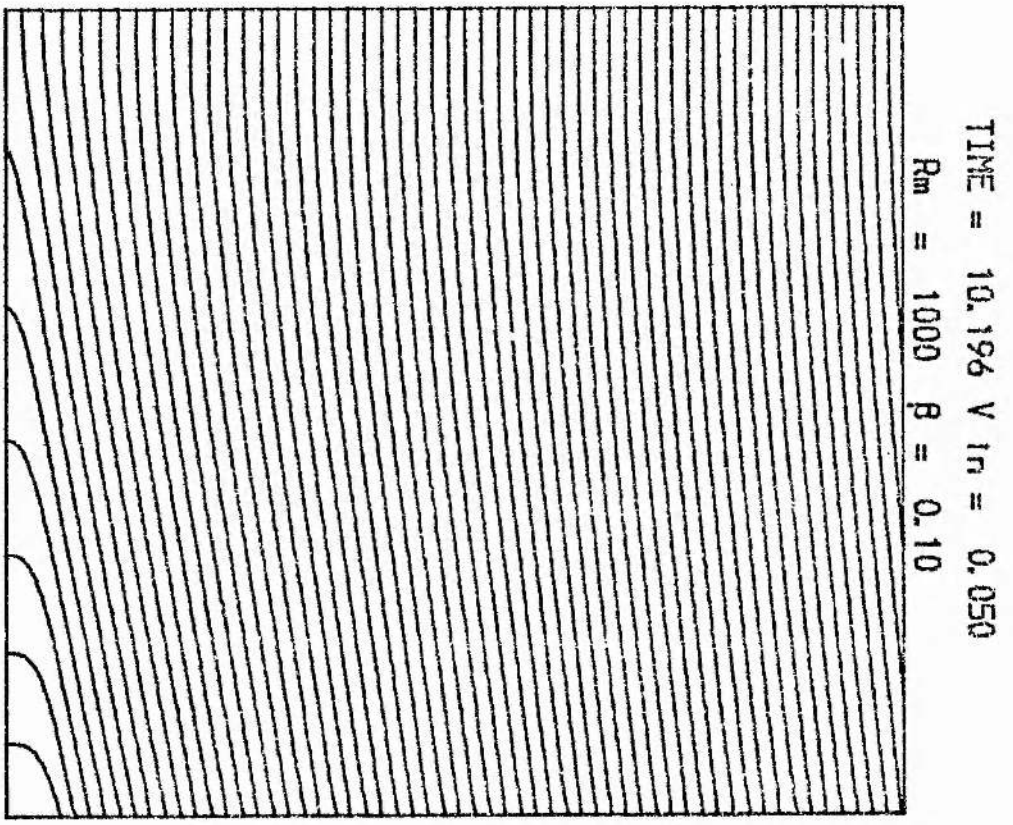


Figure 3.5 Contours of magnetic vector potential

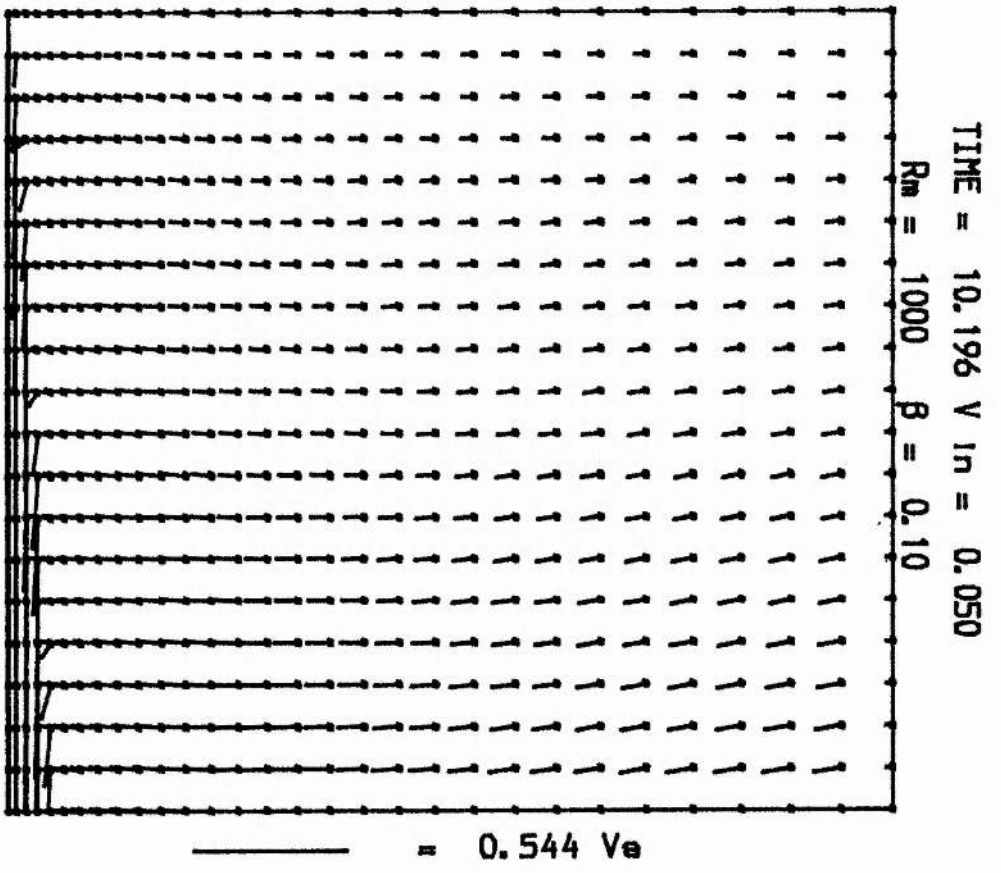


Figure 3.6 Plasma flow vectors

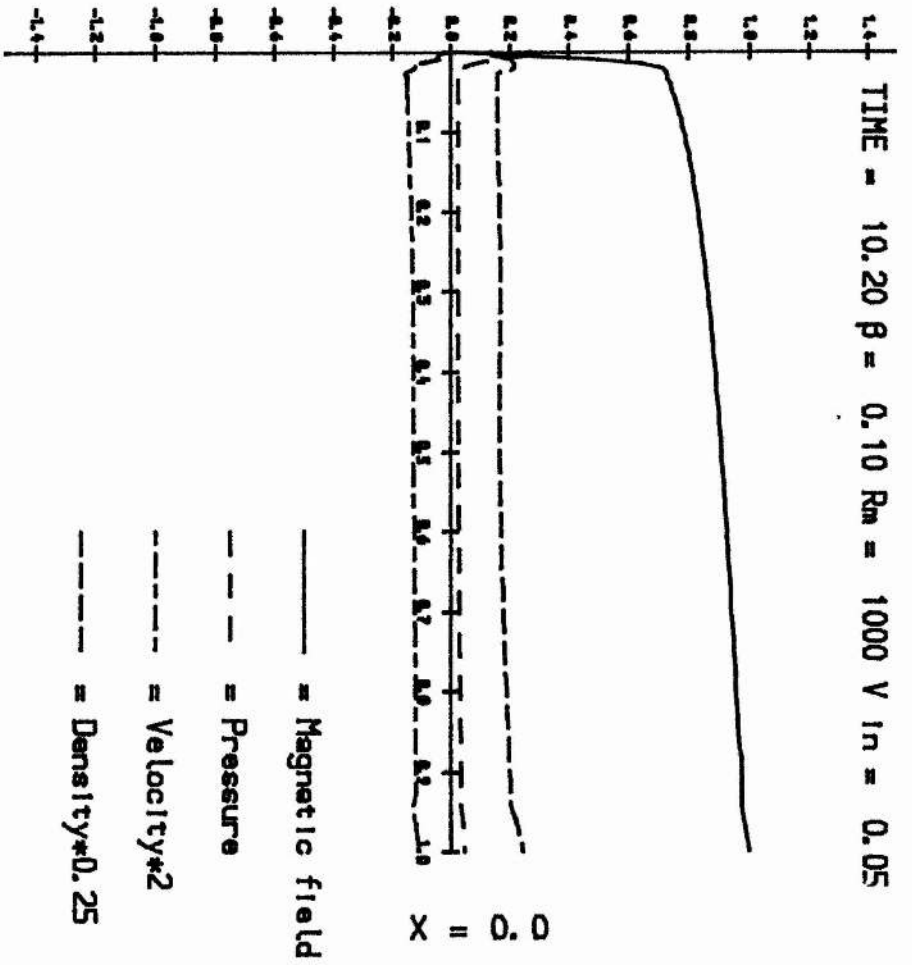


Figure 3.7 Magnetic field, pressure, velocity and density on the z-axis

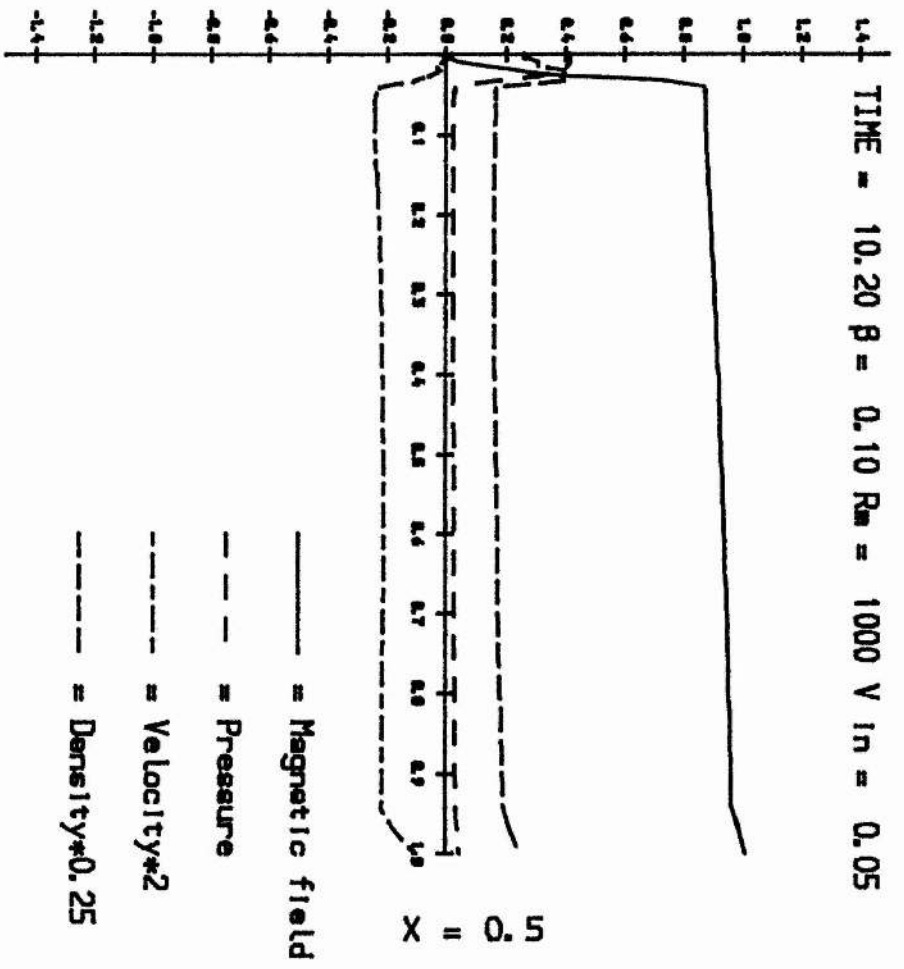


Figure 3.8 Magnetic field, pressure, velocity and density on the line x = 0.5

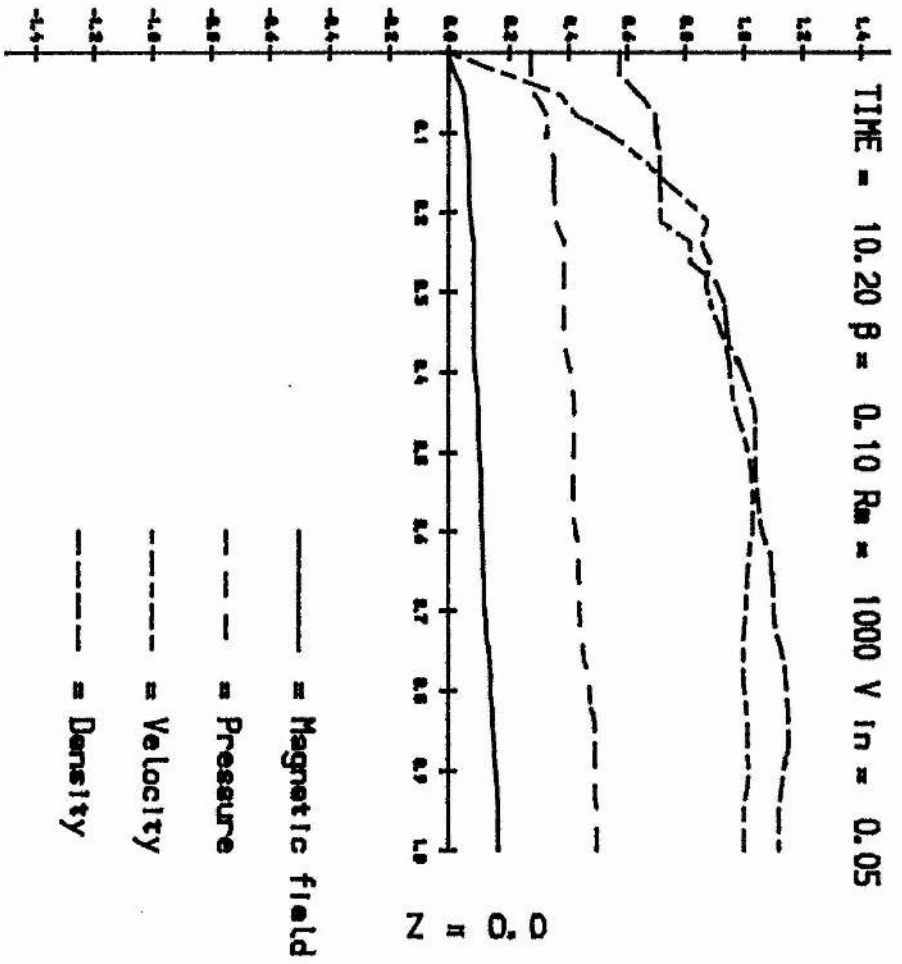


Figure 3.9 Magnetic field, pressure, velocity and density on the x-axis

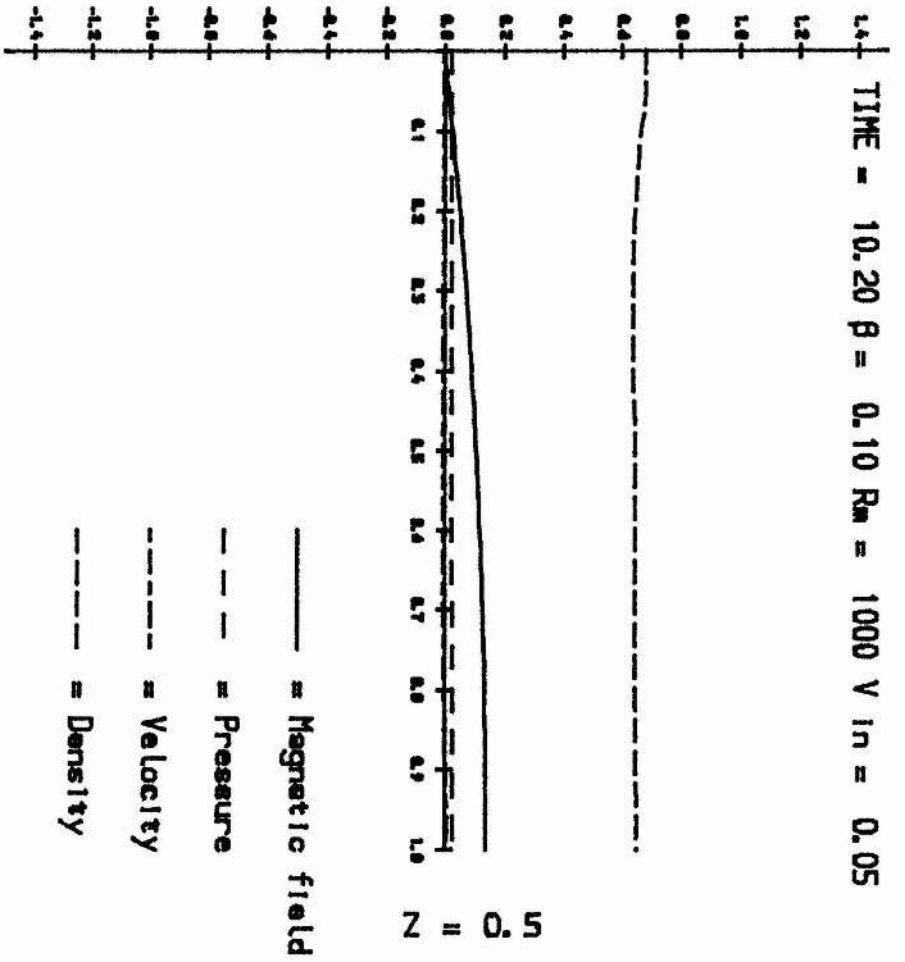


Figure 3.10 Magnetic field, pressure, velocity and density on the line z = 0.5

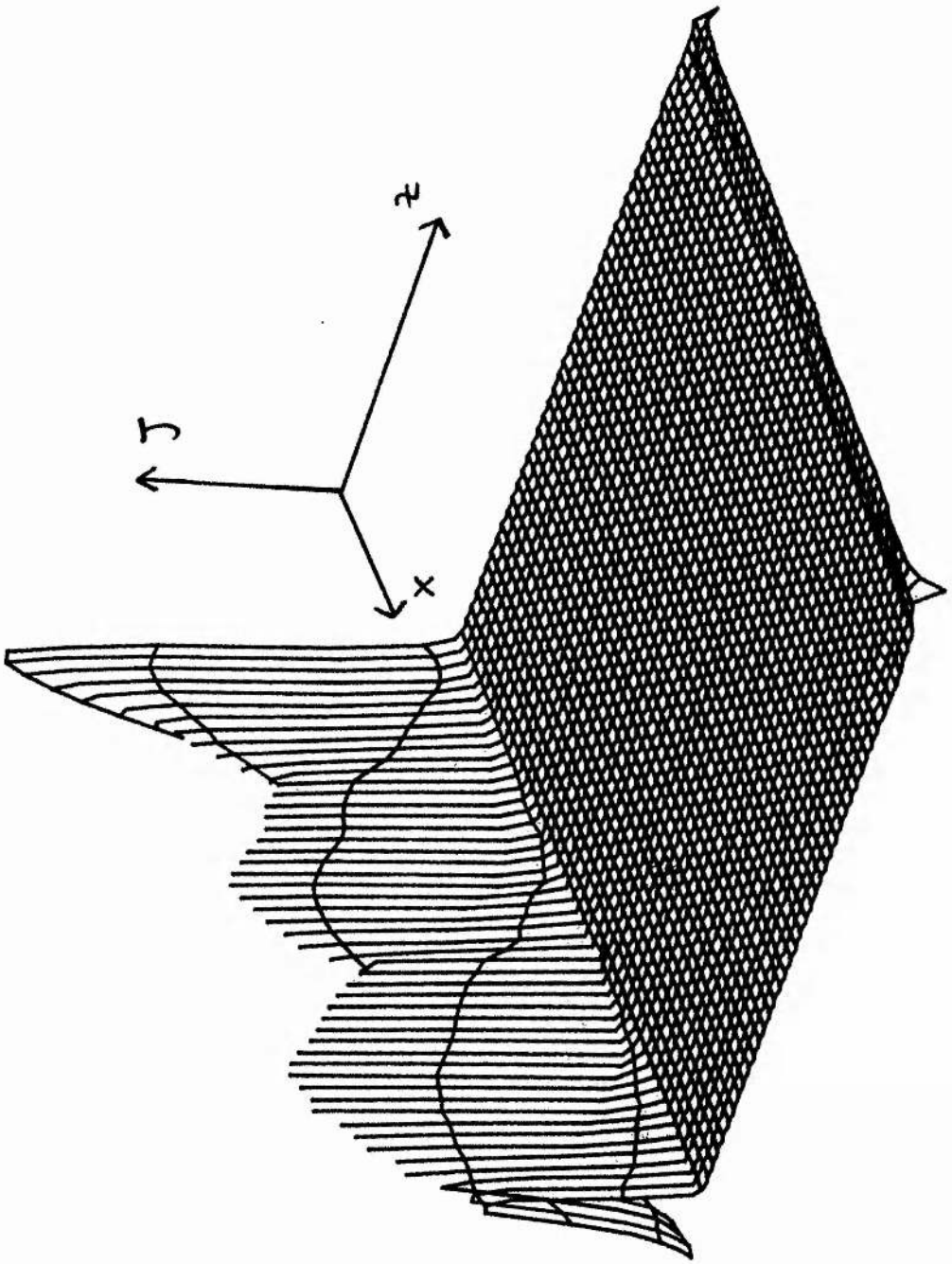


Figure 3.II Current density (J_y) plotted against x and z

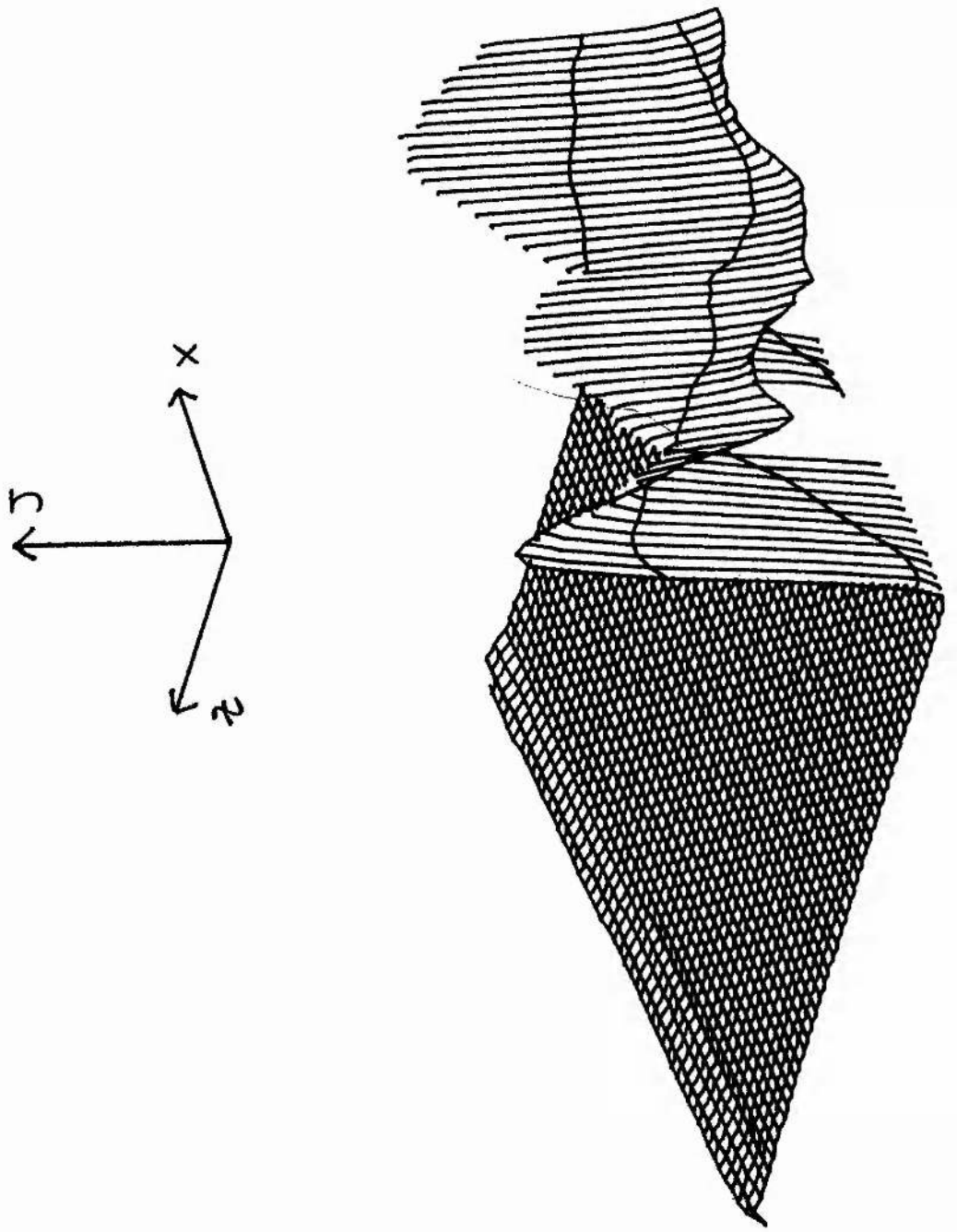


Figure 3.I2 Current density (J_y) plotted against x and z; opposite orientation to Figure 3.II

CURRENT DENSITY
PLOT NO. 1 DATE 01/28/87 TIME 10.58.52
AZIM = 310.0 ELEV = 80.0 DIST = 600

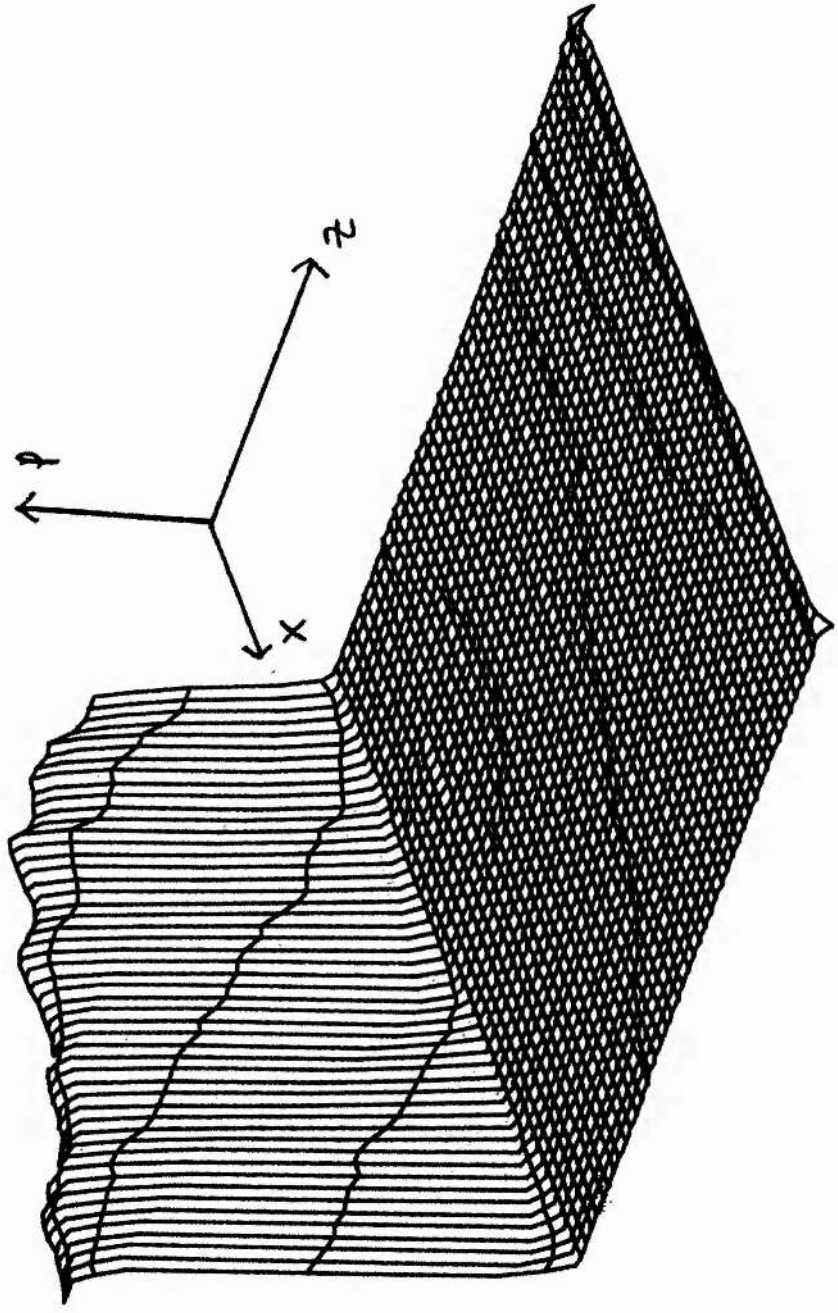


Figure 3.13 Plasma pressure plotted against x and z

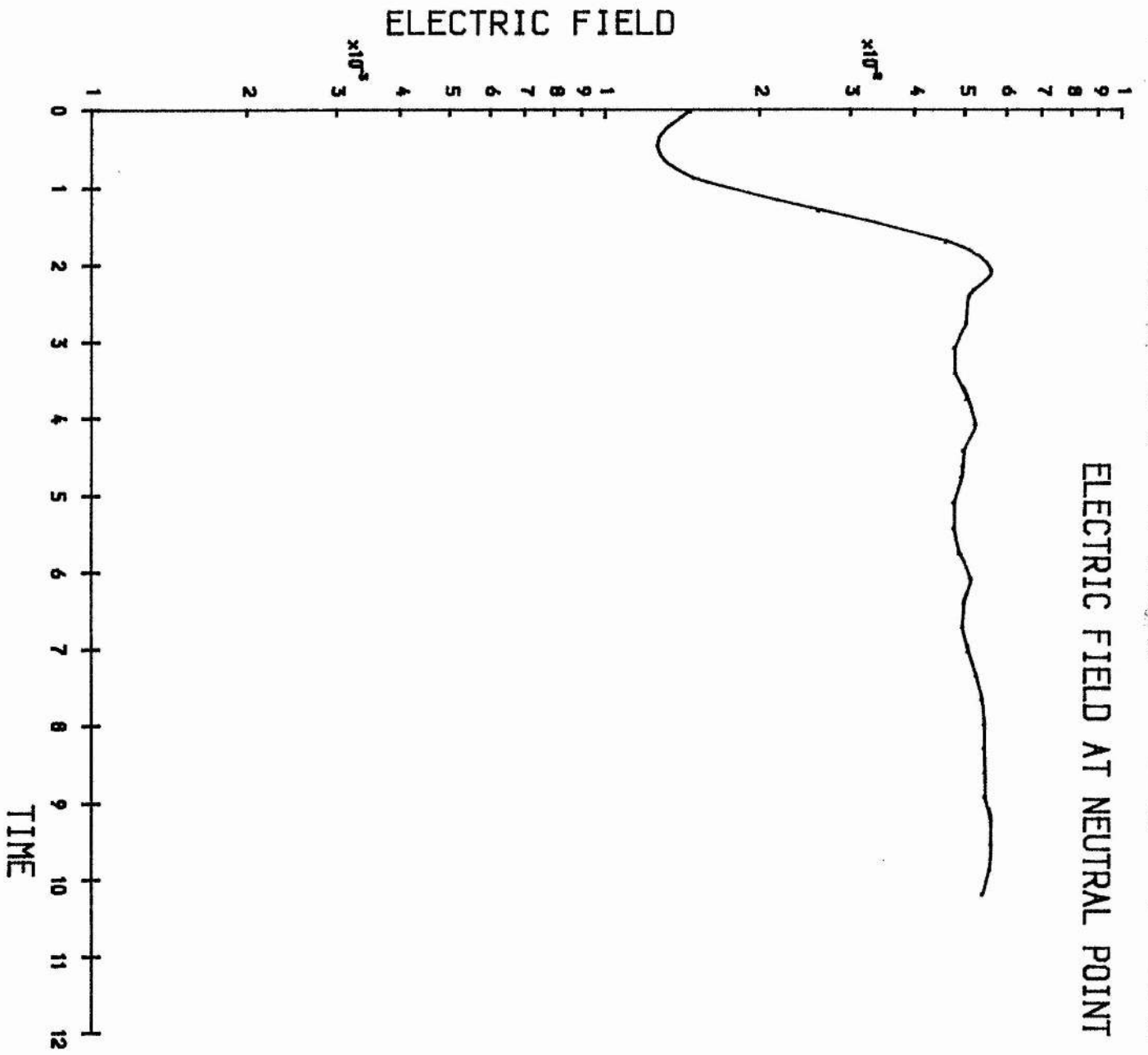


Figure 3.I4 Electric field at the neutral point (x, z = 0) as a function of time

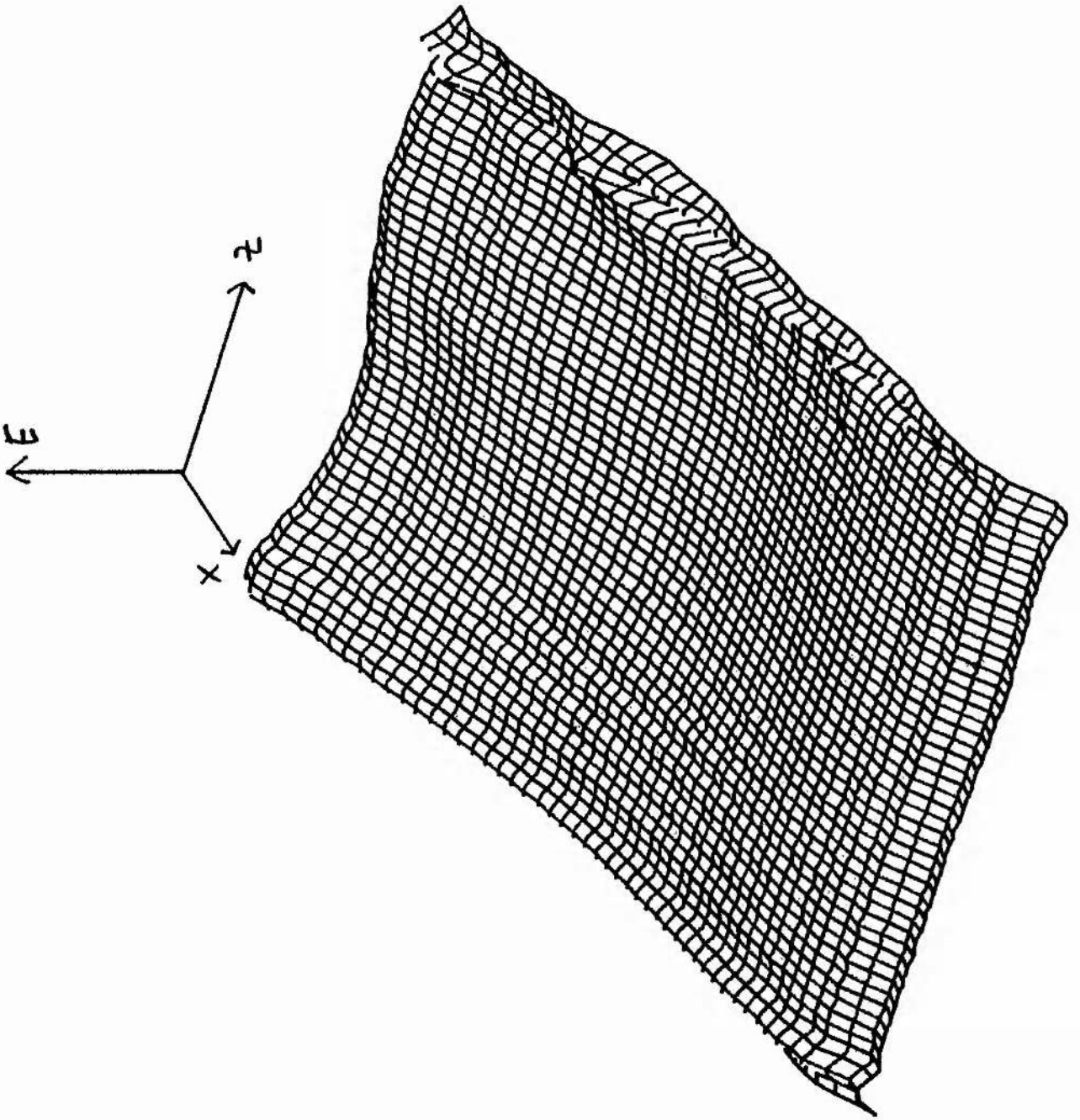


Figure 3.I5 Electric field (E_y) plotted against x and z

x and z

The plots given show all the features of a 'classical' driven reconnection model. Field lines are driven in toward the diffusion region at a few percent of the external Alfven speed, where they break and reconnect. Once a field line passes the separatrix its curvature becomes much greater, and the tension force then sweeps flux and plasma out of the box at 90 degrees to the original direction of approach. The current sheet remains in place but changes in response to the external conditions. Thus, at the neutral point where diffusion is at a maximum, the value of J has the highest value, while at the outflow boundary the sheet has bifurcated into a pair of standing slow magnetohydrodynamical shock waves (3.11, 3.12). All these features may be seen in other studies (Sato, 1979; Biskamp, 1986) although the central peak is lost in Sato's work because he assumes a region of anomalously high resistivity at the X-point.

A close examination of the structure of the current sheet shows a rather more complex structure than is normally expected. Two subsidiary peaks appear along each shock wave. Examining the structure of the current sheet in time shows that these are not waves propagating out of the box, but are stable structures connected with the shocks in some way not understood at present. We suggest they may be related to the reverse-current spikes seen at the ends of the central current sheet in Biskamp's simulations.

The corresponding three-dimensional plot of plasma pressure (3.13) shows structure along the central ridge. The variations on small length scales are not physical but arise from numerical noise generated by the

FCT algorithm. Robertson and Frank (1986) note that pressure is particularly vulnerable to the appearance of numerical noise when such an algorithm is used, and suggest that it occurs because of the tendency of adjacent cells to pair off so that the run of a variable is not smooth but stepped. This may ultimately derive from clipping of the extrema of the advected variables.

We also note that the plasma pressure increases as we pass along the current sheet, implying that the plasma acceleration occurs because of tension in the magnetic field lines rather than from hydrodynamic causes.

The graph of electric field against time (3.14) demonstrates that a steady state has been reached after two Alfven travel times. A close examination of the graph of E shows slight oscillations about the steady state with a period of about two Alfven times. We interpret these as being caused by Alfven waves propagating between the boundaries of the box. The amplitude of the waves dies down in time, probably because of energy dissipation. In Figure (3.15) we plot the electric field over the region of computation when the system has reached a steady state. While E_y is approximately constant in the internal region, the boundary conditions are having a slight effect on this variable at the edges of the computational domain.

It is interesting to compare this solution with the fast steady state reconnection models developed by Priest and Forbes (1986). A close examination of the plasma flow vectors (3.6) shows that the flow both converges and diverges in a way not consistent with any of their solutions. However, if we restrict attention the the region $z < 0.5$ some properties characteristic of a fast mode expansion (magnetic field and

pressure decreasing towards the diffusion region) are seen (3.7, 3.8). This suggests that the run lies within the hybrid expansion regime of their range of solutions.

We may quantify this statement on calculating the paramater b_0 from Priest and Forbes' 1986 theory. b_0 may be determined by examining the variation in plasma pressure on approaching the current sheet. Equation (37) from the 1986 paper of Priest and Forbes gives

$$p_0 + p_1 = p_0 - \frac{B_0}{\mu} a_0 b_0 \sinh\left[\frac{\pi}{2}(1-y)\right] \qquad (3.3.1)$$

where p_0 is the unperturbed pressure far away from the current sheet, p_1 is the first order correction to the plasma pressure and B_0 is the unperturbed magnetic field (equal to one in our non-dimensionalised units). a_0 is given by their equation (35):

$$a_0 = \frac{16 B_N \sin(\pi L/2Le)}{(L/Le)\pi^2 \cosh(\pi/2)} \qquad (3.3.2)$$

where $B_N = B_\infty \times M_\infty$, L is the length of the central diffusion region and L_∞ is the length scale of the system, which is also one in our units. Substituting the measured values at (0.0, 0.1) and (0.0, 0.8) into these expressions gives a value for b_0 of 0.11.

3.3 Effect of varying the magnetic Reynolds number

We define a magnetic Reynolds number

$$R_m = L\, v_A / \eta$$

where L refers to the global length scale of the system, v_A is the Alfven speed and η is the non-dimensional resistivity of the plasma.

Five runs of the code were performed in which Rm was varied by changing η and keeping all other externally adjustable parameters constant after the initial transient phase (Figures 3.16 - 3.19). From Faraday's law, the electric field will be constant over the entire computational domain when the system has reached a steady state. Our simplified Ohm's law

$$\underline{J} = \frac{1}{\eta} (\underline{E} + \underline{v} \times \underline{B})$$

then implies that $\underline{v} \times \underline{B}$ in the current-free external region will have the same value as $\eta \underline{J}$ at the neutral point, which gives the rate of reconnection of the system.

We may now discuss the formation of the boundary layer near the inflow region. If it were possible to impose the electric field at the inflow boundary through $\underline{B}$ and $\underline{v}$, we would also be imposing E at the neutral point, so that the graphs of E for different runs (Figure 3.20) should all show the same values. Since this is not the case, we suggest instead that for every η there is a natural unique reconnection rate that imposes an associated electric field over the box. The value of this electric field may be different from that given by the boundary conditions. This would not be possible for a steady-state current-free field, and in fact the magnetic field and flow patterns are readjusting on passing into the computational domain. This gives rise to the boundary layer seen on cross-sections of the system near the inflow boundary, together with the current ridge we saw near the inflow region in most runs. These jumps are more apparent in v than B because of the low plasma β; the magnetic field dominates the hydrodynamic effects.

The probable cause of this boundary layer is overspecification of the inflow variables as discussed in Forbes and Priest (1987). This effect is probably present in all runs and is closely related to the

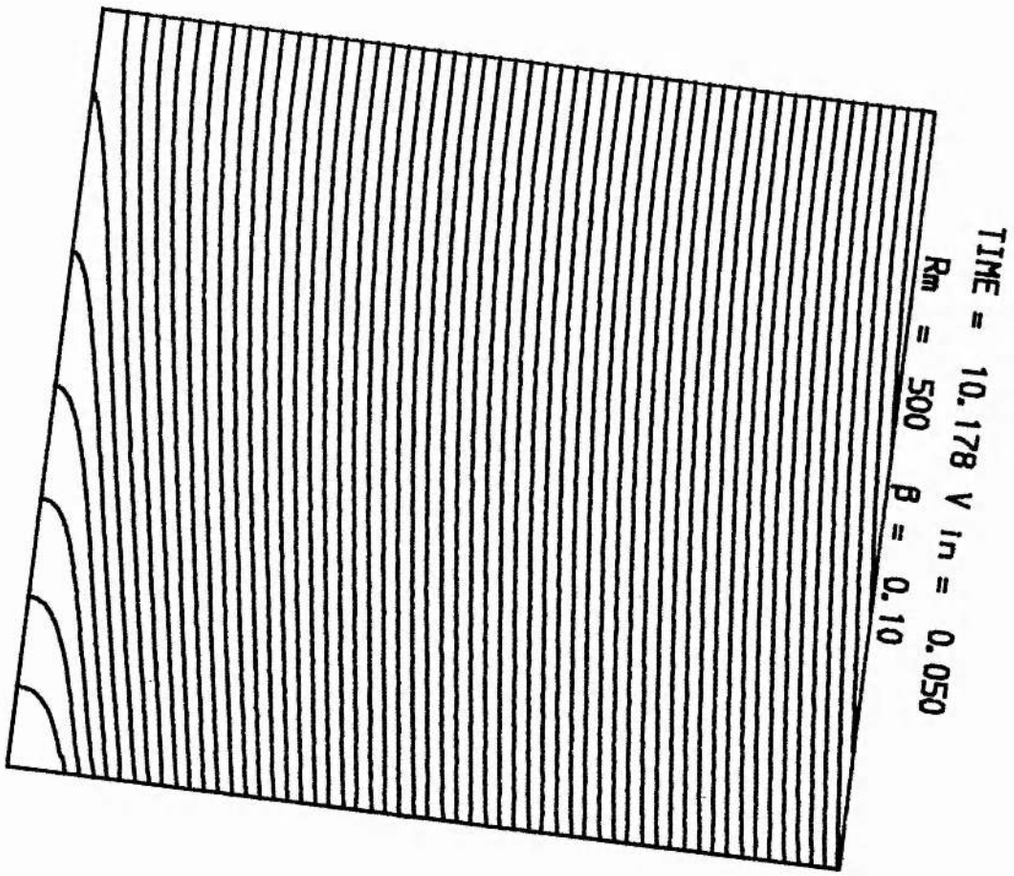


Figure 3.16 Contours of magnetic vector potential: $R_m = 500$

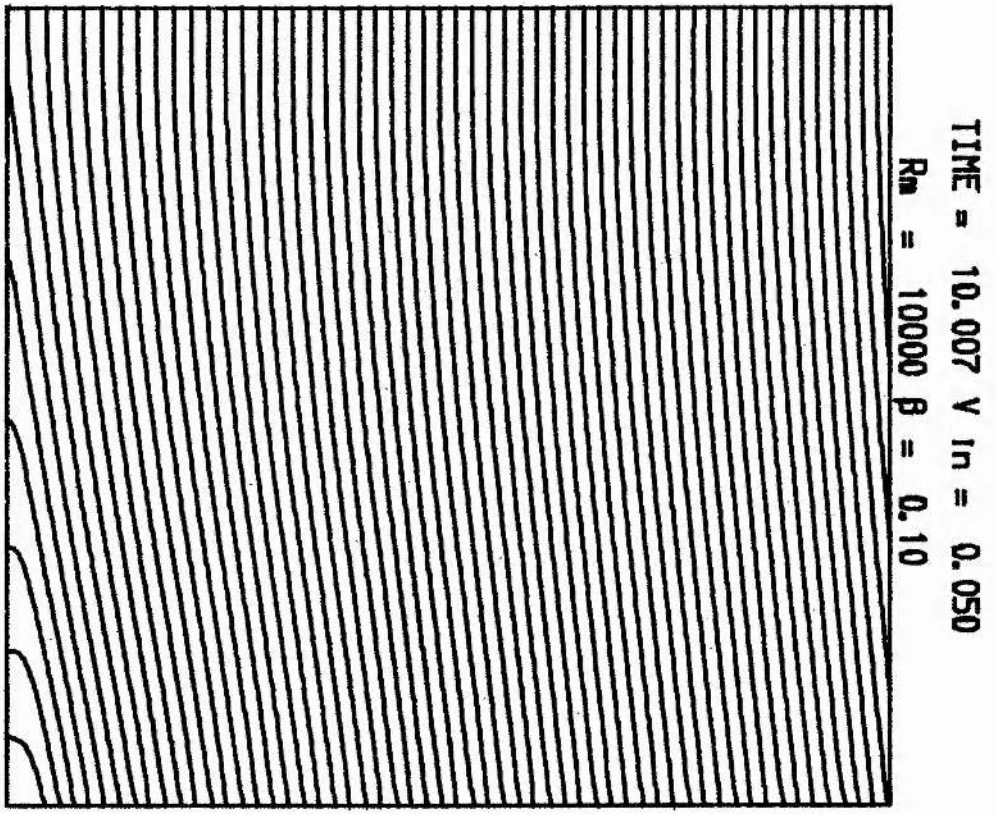


Figure 3.I7 Contours of magnetic vector potential: R_m = I0000

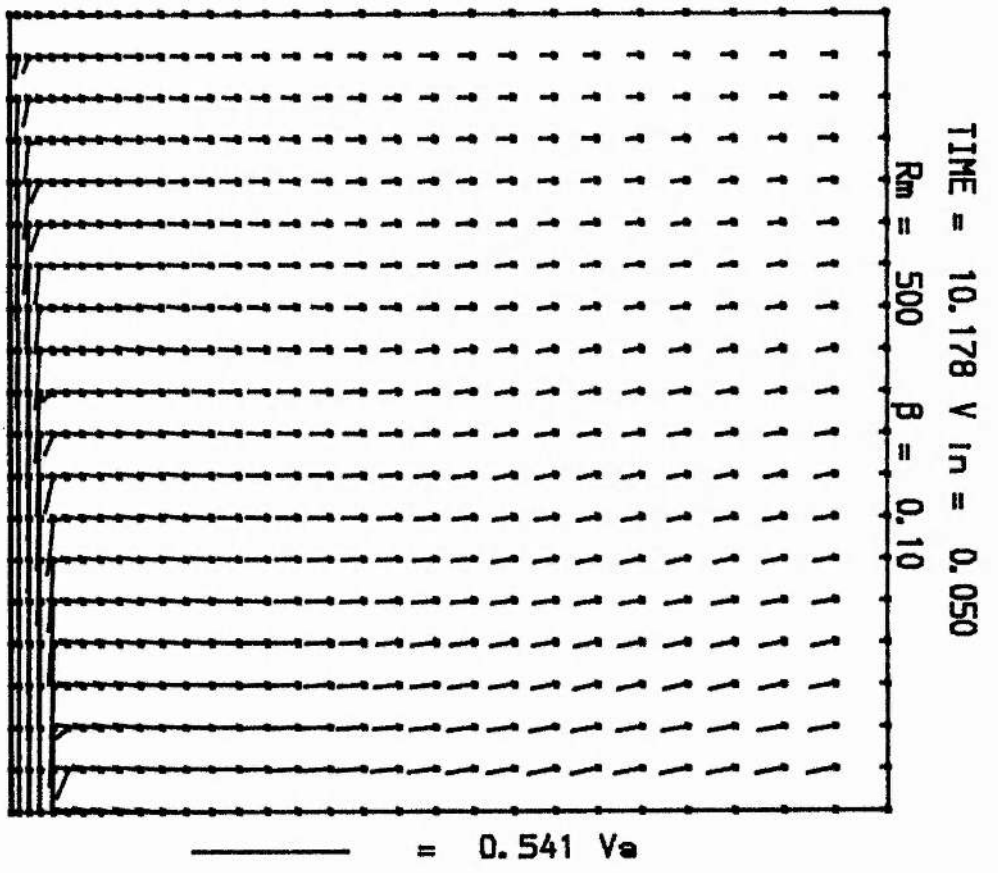


Figure 3.I8 Plasma flow vectors: R_m = 500

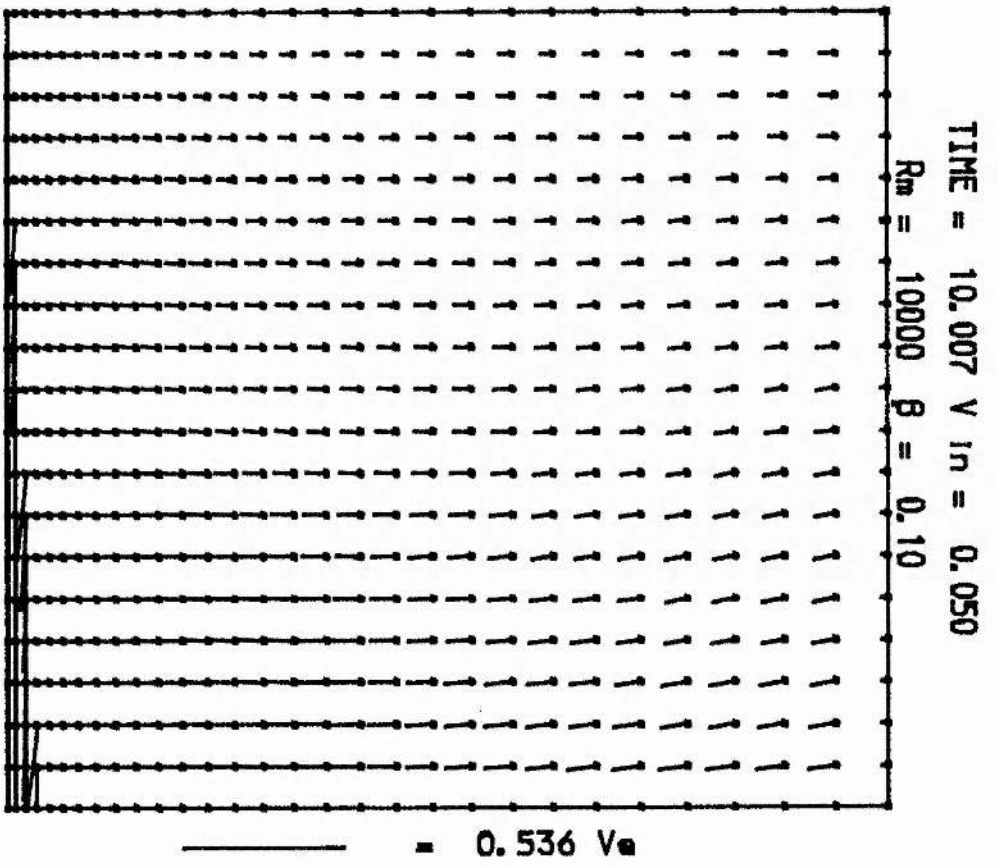


Figure 3.I9 Plasma flow vectors: R_m = I0000

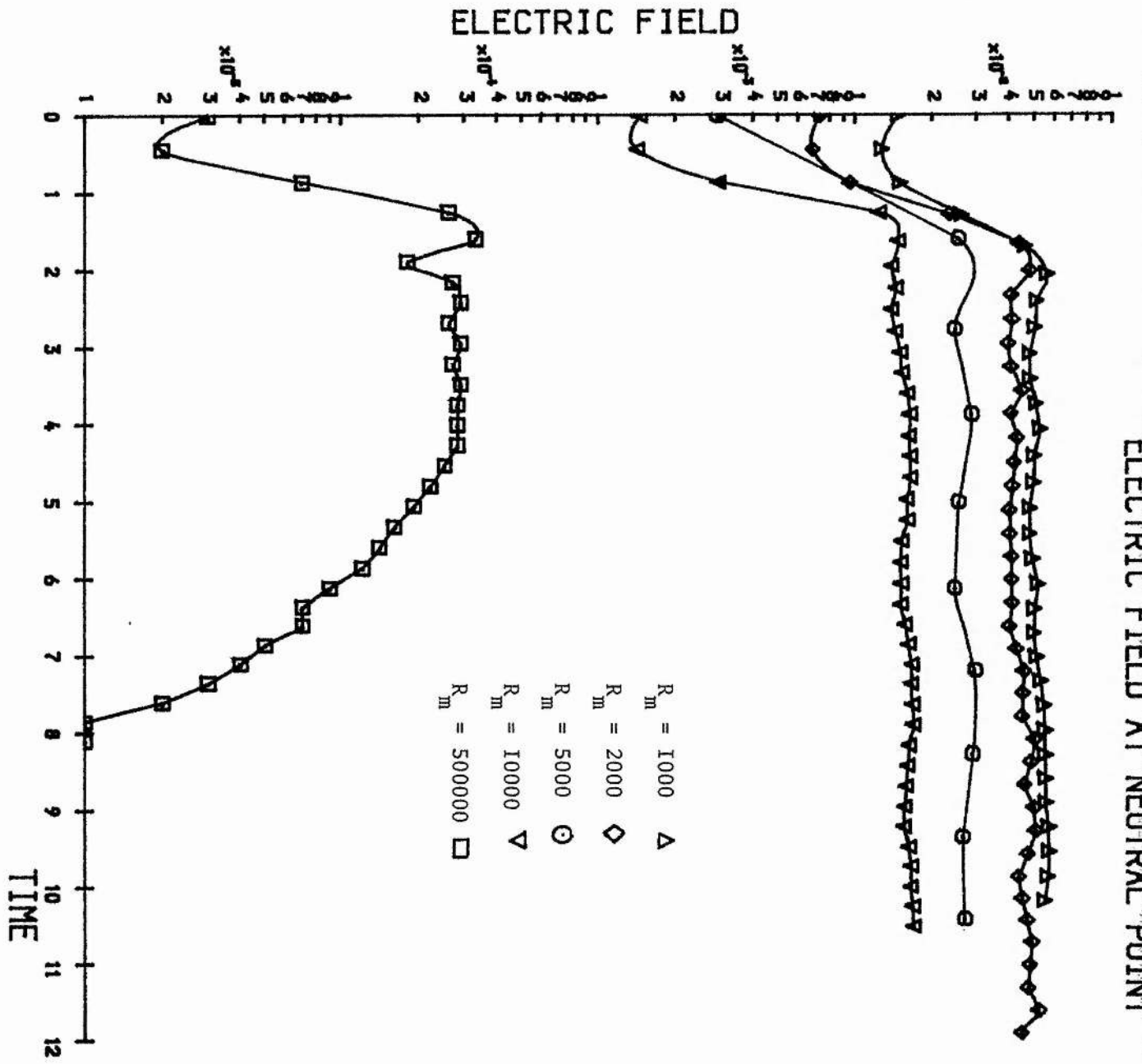


Figure 3.20 Electric field at neutral point for different R_m

electric field mismatch. Further experiments, which we describe below, show that keeping the magnetic Reynolds number constant while varying α and the plasma inflow speed does not substantially affect the reconnection rate.

The global flow patterns are clearly more complex than any theory has so far predicted. The solution appears to be dividing itself into two regions which we may separate for convenience by the line $z = 0.5$. In the upper region, the vectors diverge away from the z axis in response to the curved magnetic field imposed at the boundary. In contrast, the flow in the lower region is dominated by the reconnection. A quasi-Petschek flow has formed with an associated weak slow-mode expansion, suggesting that these solutions lie within the hybrid regime of Priest and Forbes' classification scheme

3.4.1 Size of the diffusion region

We shall define here the edge of the diffusion region to be the contour about the X-point at which the η J and vB terms in Ohm's law are equal in magnitude. In Table 3.1, L and l denote lengths of the diffusion region in the x and z directions respectively.

TABLE 3.1 $\alpha = 0.10$ Inflow speed = 0.05

Rm	L	l	L/l
500	0.126	0.024	5.2
1000	0.107	0.013	8.3
2000	0.053	0.007	7.7
5000	0.080	0.002	[42]
10000	0.034	0.001	[48]

L and l are calculated by linear interpolation between gridpoints. The first three runs have the width l of the current sheet lying between three and seven gridpoints and the diffusion region will be adequately resolved. For the last two runs, the diffusion region is being modelled by one gridpoint only in the z direction, so the numbers quoted in square brackets are not as accurate as those in the upper part of the table. However, the aspect ratio L/l appears to grow as η is increased, a result also noted by Biskamp (1986).

It is of interest to examine the scaling of l with respect to the magnetic Reynolds number (Figure 3.21), as do Biskamp (1986) and Robertson and Priest (1987). The scaling law derived from this graph suggests that

$$\ell \sim R_m^{-1}$$

so our results differ from both Robertson and Priest's scaling law

$$\ell \sim R_m^{-\frac{1}{2}}$$

and Biskamp's

$$\ell \sim R_m^{0}$$

Biskamp suggests that if the aspect ratio becomes too high the current sheet will become tearing unstable. This is borne out both by his simulations (he finds the central current sheet does tear and the resulting islands are swept downstream and out of the numerical box) and by the run shown at Rm = 500000 (Figures 3.22, 3.23, 3.24).

This final run is included for interest, for the resolution of the current sheet at the neutral point will probably not resolve all length

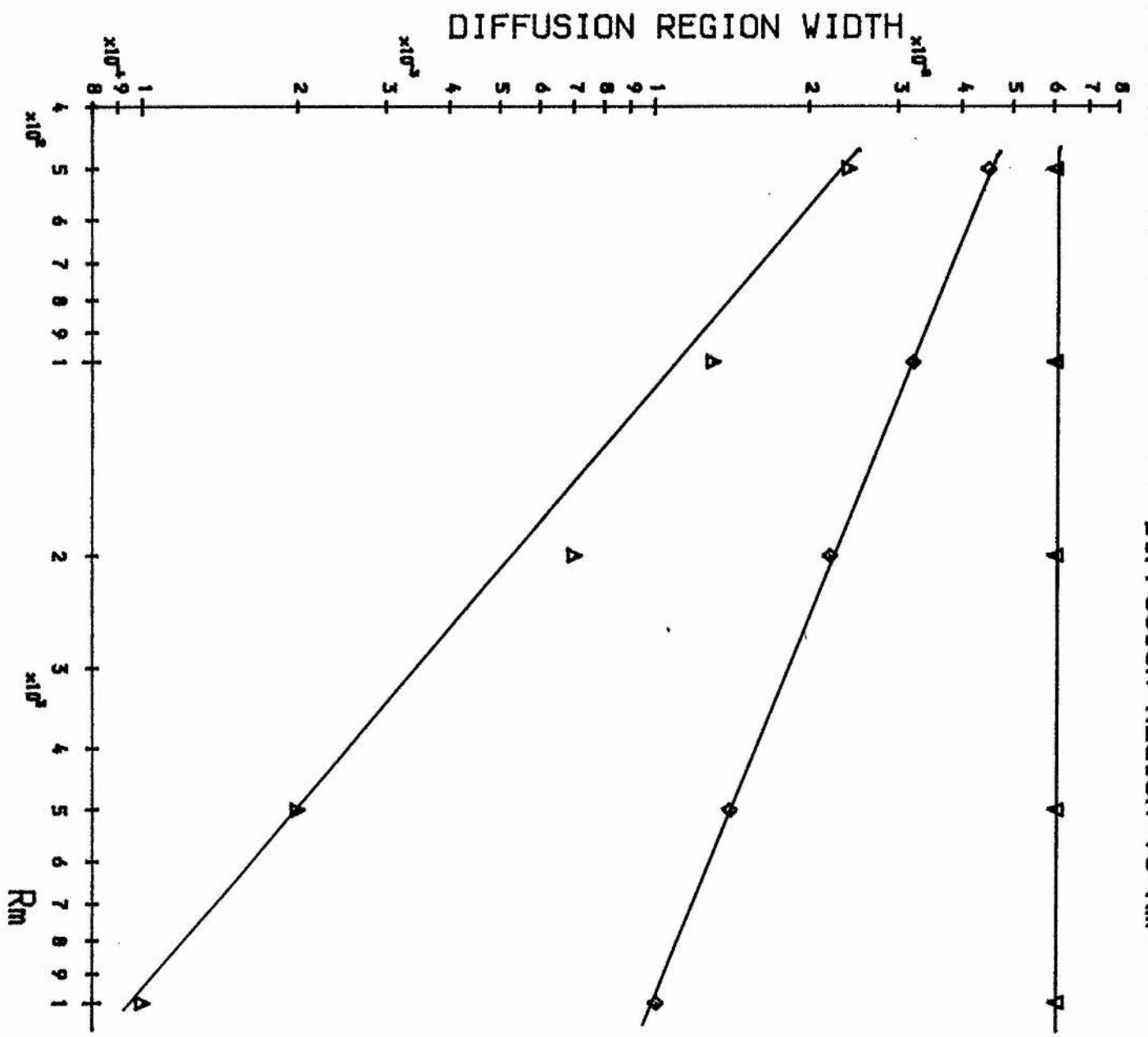


Figure 3.2I Width of diffusion region against R_m

∇ : Prediction from Biskamp's scaling law

$\Diamond$: Prediction from Robertson and Priest's scaling law

Δ : Experimentally determined scaling

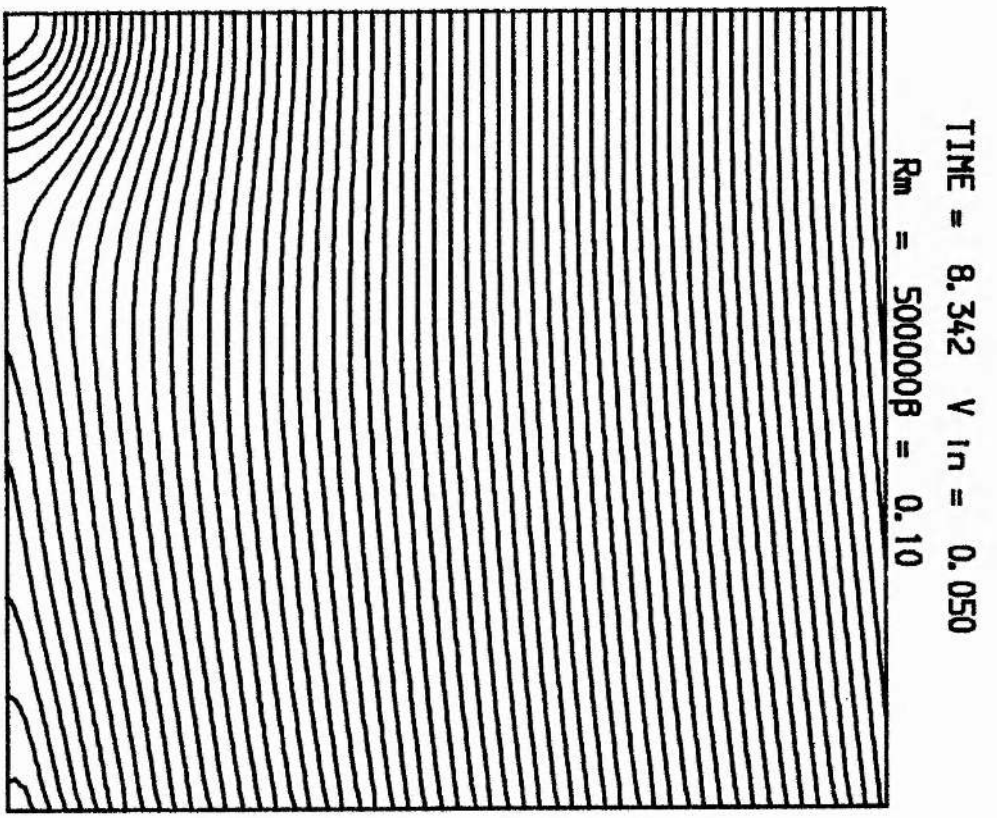


Figure 3.22 Contours of magnetic vector potential; R_m = 500000

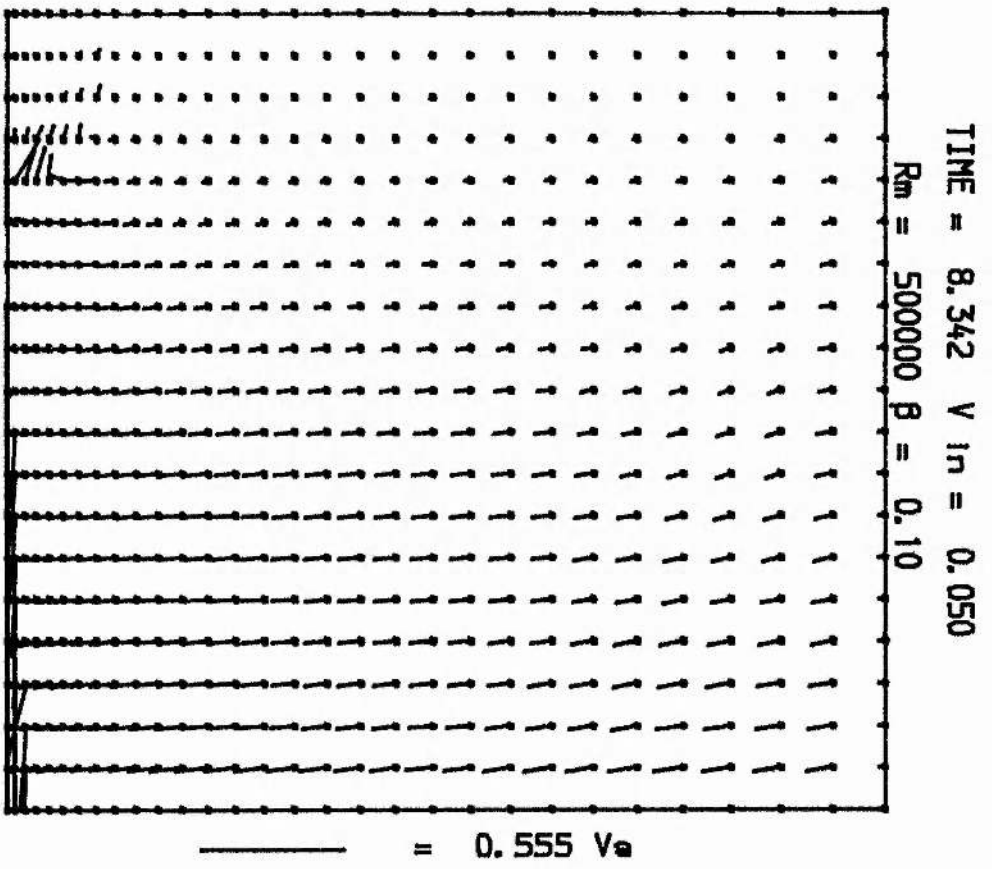


Figure 3.23 Plasma flow vectors; R_m = 500000

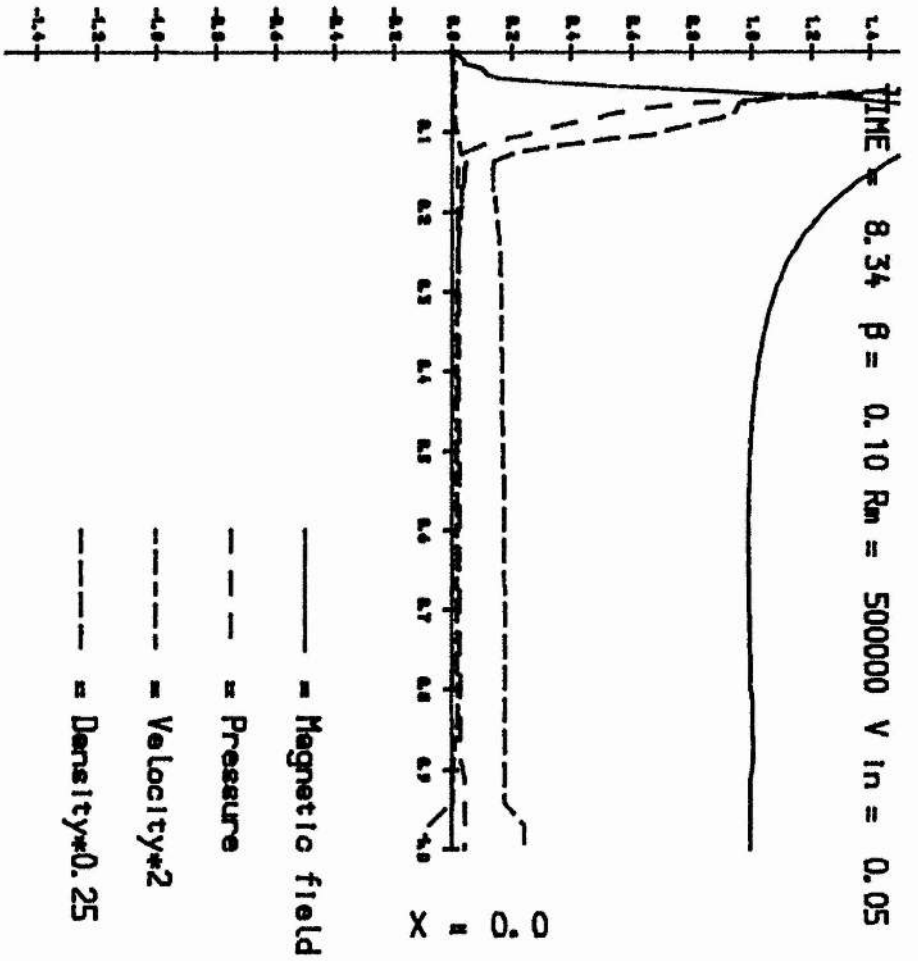


Figure 3.24 Magnetic field, pressure, velocity and density along the z-axis; R_m = 500000

scales. However, the qualitative behaviour shown by the code agrees with both Biskamp's findings and the theoretical predictions. This was the only run performed in the study for which B_x was found to increase on approaching the current sheet.

The plot of electric field at the neutral point (Figure 3.20) shows the three stages of development when Rm = 500000. In the 'rise' stage the inflow variables are being linearly increased to their full values, with a corresponding rise in electric field. In the 'plateau' stage the electric field has stabilised but the current sheet is increasing in length because of flux pile-up (this run was the only occasion we saw any such behaviour). Lastly, in the 'tearing' stage the aspect ratio of the current sheet exceeds some critical value and a magnetic island forms at the neutral point. We then see a complex mixture of flow patterns, with a clear tearing mode at the neutral point and a jet of accelerated plasma further along the x axis. Theoretical calculation of the exact critical aspect ratio will probably be extremely difficult, as a tearing mode analysis will have to be repeated with a time-dependent external flow, magnetic field and length scale.

3.4.2 Alfven Mach numbers

We compare the measured local Alfven Mach numbers in the plasma with the predicted values from the Sweet-Parker and Petschek theories. A scale length of unity is assumed for the magnetic Reynolds numbers used to derive the theoretical reconnection rates, described in Chapter 1, in table 3.2.

Table 3.2 Alfven Mach numbers α = 0.1 Inflow speed = 0.05

Rm	M_e	M'_e	M_i	M_{SP}	$M_{Pet\,i}$	$M_{Pet\,e}$	E
500	0.060	0.104	0.079	0.045	0.046	0.063	0.060
1000	0.055	0.099	0.072	0.032	0.042	0.057	0.054
2000	0.041	0.101	0.064	0.022	0.036	0.052	0.050
5000	0.064	0.103	0.063	0.014	0.034	0.046	0.026
10000	0.062	0.106	0.051	0.010	0.031	0.043	0.016

where M_e, M'_e, M_i are the Alfven Mach numbers measured with respect to the local Alfven speed at (0.0,0.8), (0.5,0.8), (0.0,0.1) respectively. M_{SP} is the Sweet-Parker reconnection rate and M_{iPet}, M_{ePet} are the predictions for the internal and external Mach numbers from Soward and Priest's extension to Petschek's theory, for which expressions are given in Chapter 1. Estimates for M_{iPet} are based on the length of the diffusion region as given in table 3.1, while E is the electric field measured at the neutral point.

Some general trends may be deduced. Firstly, the Sweet-Parker prediction is always too low. Secondly, the internally measured Mach number is scaling linearly as the Petschek prediction rather than the Sweet-Parker diffusion rate (Figure 3.25). This is again in contrast to Biskamp's results. Thirdly, the measured Mach number M'_e in the inflow region always exceeds Petschek's prediction. Apart from the first case (where diffusive effects will be most dominant) M'_e exceeds even the Sweet-Parker prediction, in accordance with our account of the formation of the inflow boundary layer. Fourthly, an examination of electric field and magnetic Reynolds number for these runs gives an estimate for the

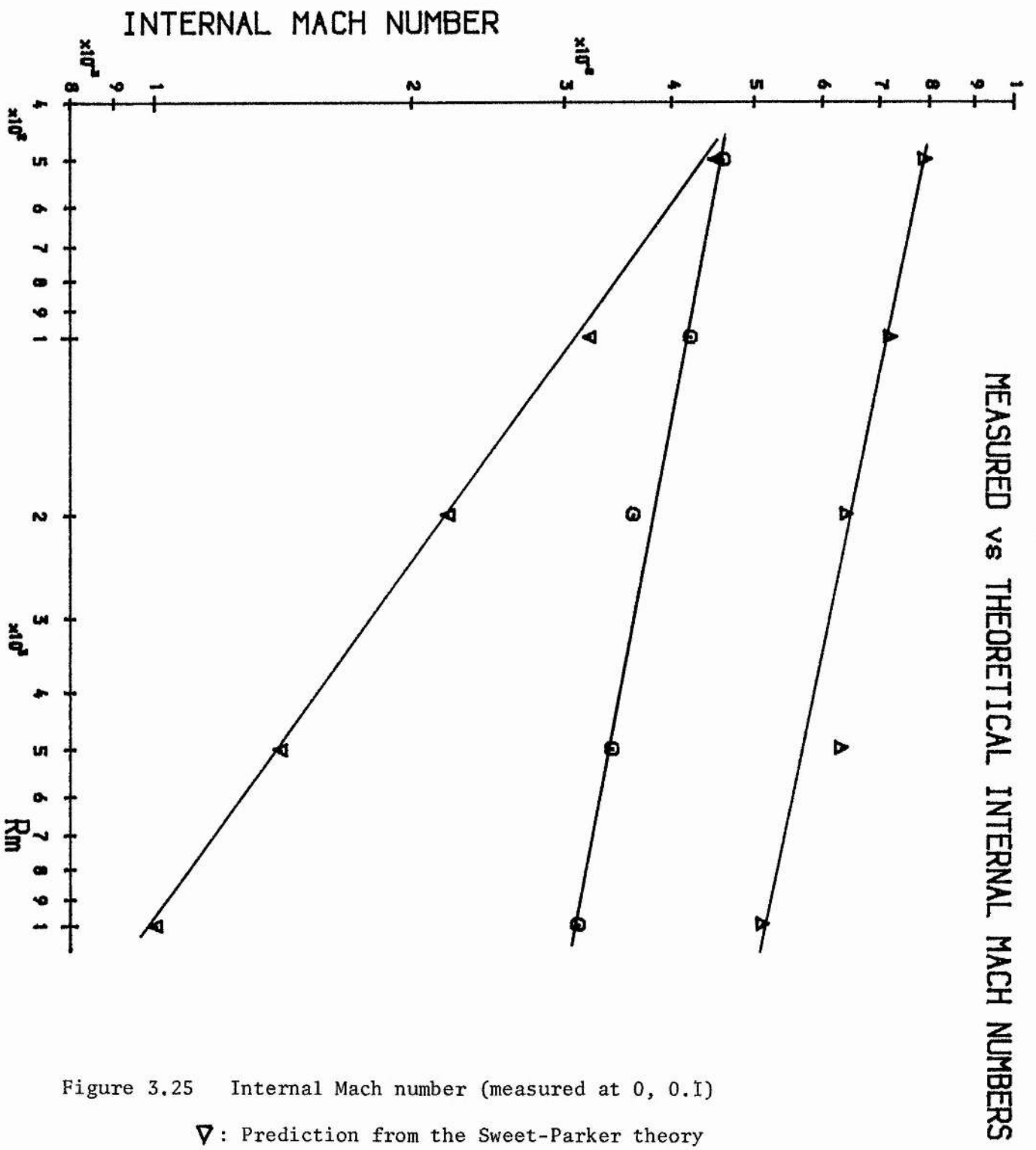


Figure 3.25 Internal Mach number (measured at 0, 0.I)

▽ : Prediction from the Sweet-Parker theory

⊙ : Prediction from the Soward-Priest theory

Δ : Experimentally determined Mach number

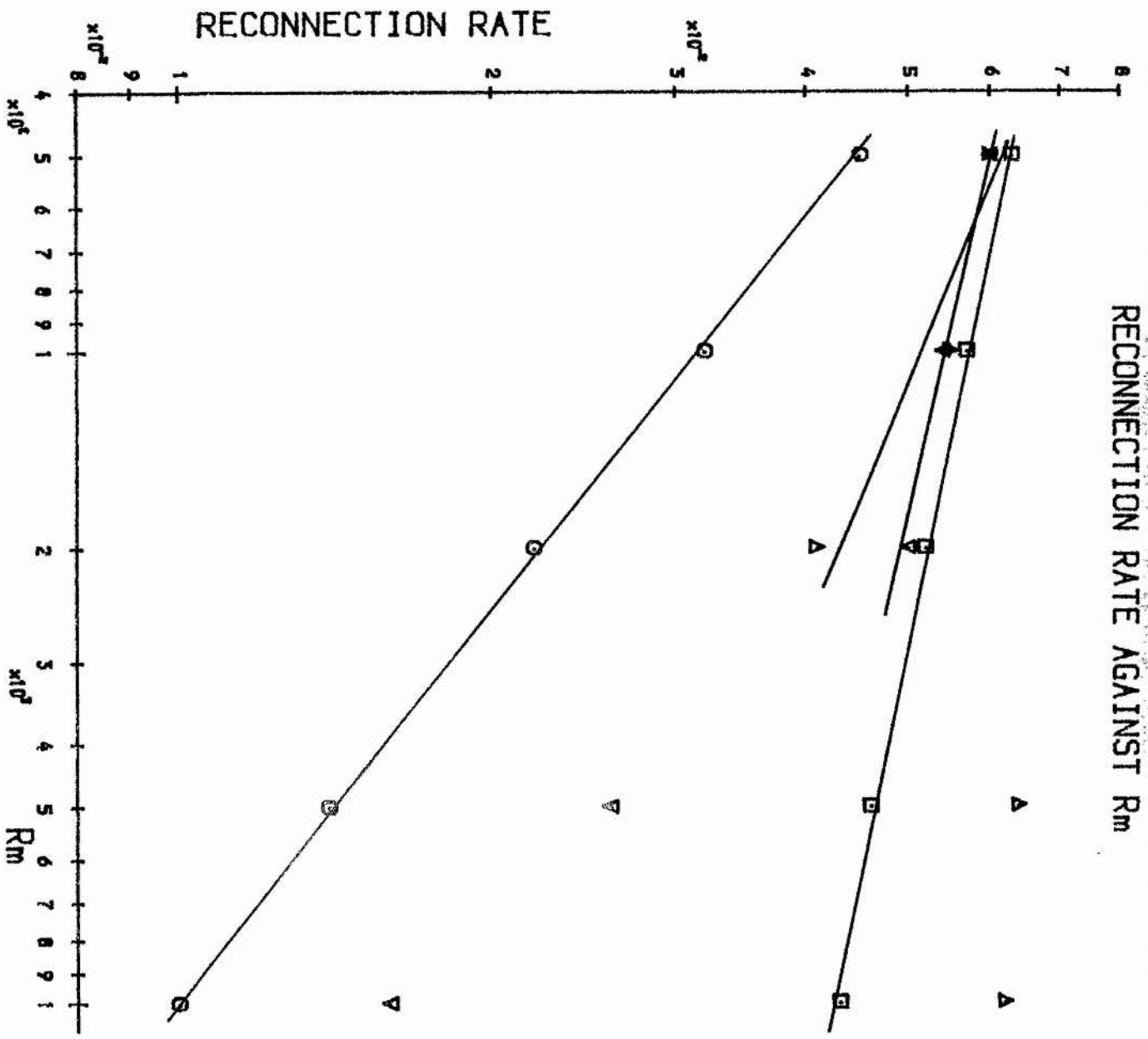


Figure 3.26 Reconnection rate and measured external Mach number against R_m

⊙ : Prediction from the Sweet-Parker theory

□ : Prediction from the Soward-Priest theory

▽ : Experimentally determined reconnection rate

△ : Mach number measured in the external region (at 0, 0.8)

rate of reconnection of the system (Figure 3.26); the expression given by Soward and Priest for M_{APet} is scaling as the experimental results.

In a steady state the rate of reconnection scales as the external inflow Mach number, so we have plotted this quantity for comparison as well. While the first three points (up to Rm = 5000) confirm the Petschek scaling, the last two do not. This may be due to insufficient resolution at the current sheet.

3.4.1 Variation of shock angles

We approximate the position of the standing slow shock wave dividing the inflow and outflow regions by a straight line between the end of the diffusion region and the point on the outflow boundary at which the x-component of the plasma velocity exceeds 0.5 of the external Alfven speed. The angle, in radians, which the shock makes with the x-axis is denoted by θ.

Table 3.3 Shock angles $\alpha = 0.1$ Inflow speed = 0.05

Rm	θ (radians)
500	0.067
1000	0.057
2000	0.045
5000	0.040
10000	0.038

The smaller η is the smaller the shock angle becomes, which again suggests that the internal behaviour of a reconnecting system is strongly dependent on this parameter.

3.5 Variation of field line curvature

A second set of runs was peformed in which the curvature of the incoming field lines α was varied systematically between zero and 0.3, while keeping η and the plasma inflow speed constant (Figures (3.27 - 3.34).

The first effect we observed was that increasing α from the value taken in the 'standard run' does not appear to alter the rate of reconnection (Figure 3.35), although the code does take significantly longer to reach a steady state. We found an upper practical limit on α of around 0.35; above this value, the discontinuity in B_z on entering the box drove the code unstable. It appears that the code could not resolve the abrupt gradients in A and B and that truncation error caused a violation of the zero-divergence condition on the magnetic field.

However, α does have a major effect on the global plasma flow pattern. While some convergence of the flow vectors appears for $\alpha < 0.1$ (Figure 3.32), the flow is divergent almost everywhere for higher α (Figure 3.28). For highly curved field lines the constant-E condition imposes an associated high tangential velocity onto the inflow vectors, which then dominates any convergence caused by the reconnection mechanism. Note that this is not flux pile-up in the sense of the Priest and Forbes models, which exhibits a strong slow-mode expansion, since the magnetic field does not increase on approaching the current sheet (Figures 3.29-3.30, 3.33-3.34).

We performed one run of the code with zero field curvature, which did not produce a non-zero Bz component, despite the non-uniform inflow velocity profile. This confirms our finding from the development stage

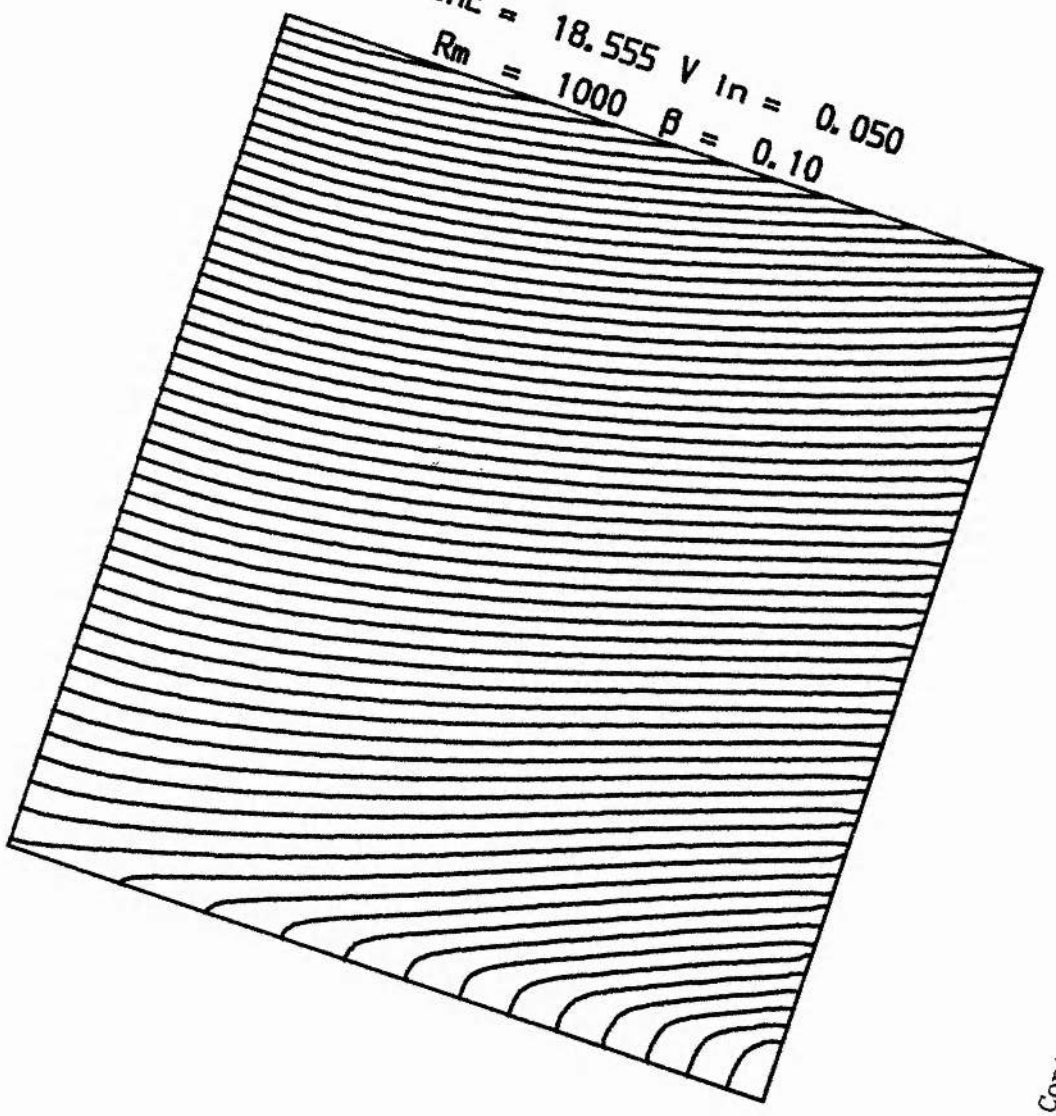


Figure 3.27 Contours of magnetic vector potential; alpha = 0.3

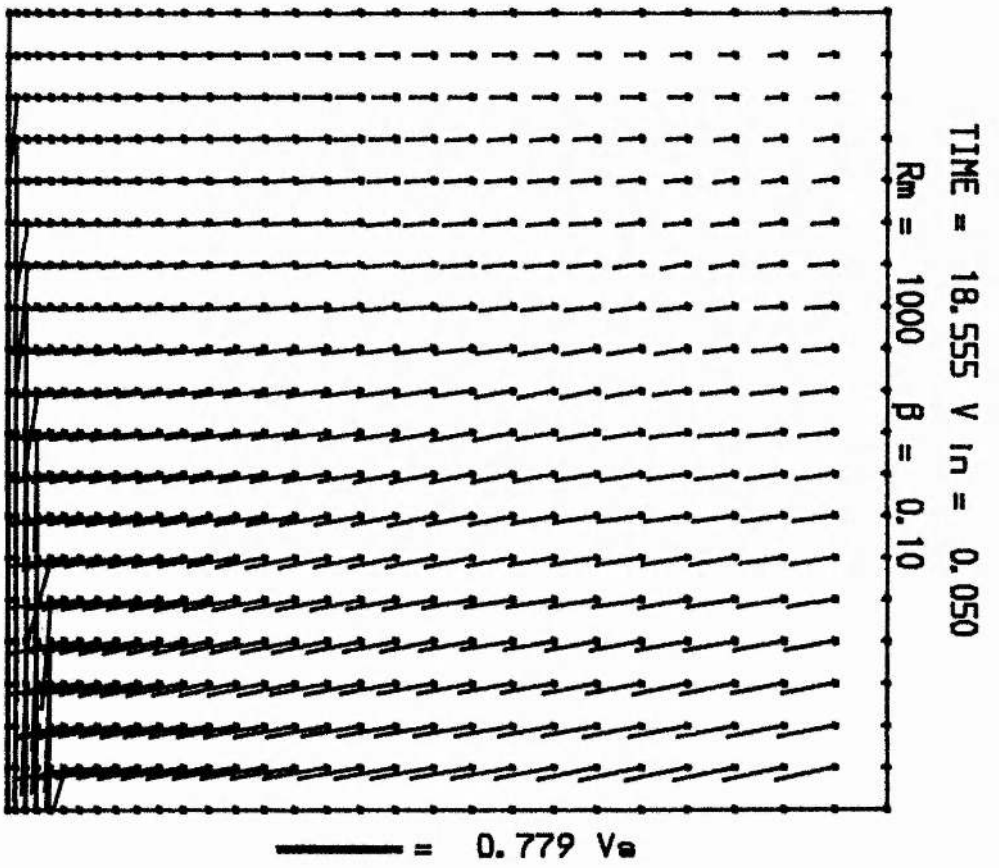


Figure 3.28 Plasma flow vectors; alpha = 0.3

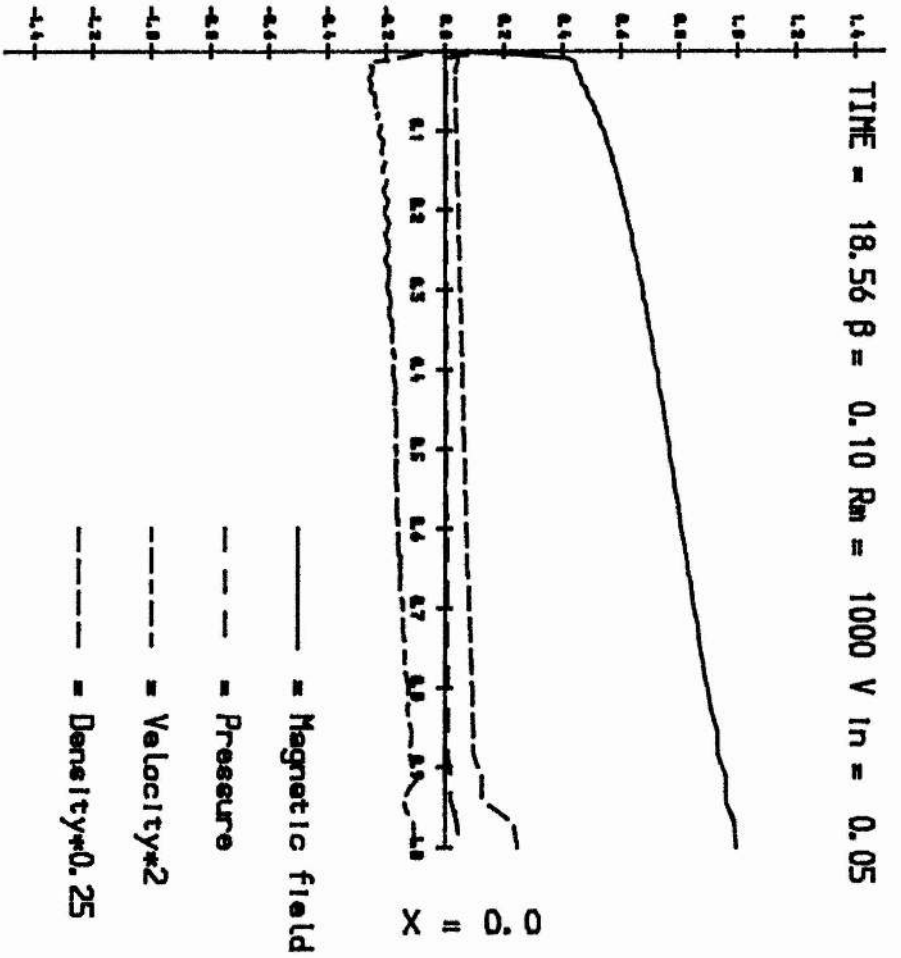


Figure 3.29 Magnetic field, pressure, velocity and density on the z-axis; alpha = 0.3

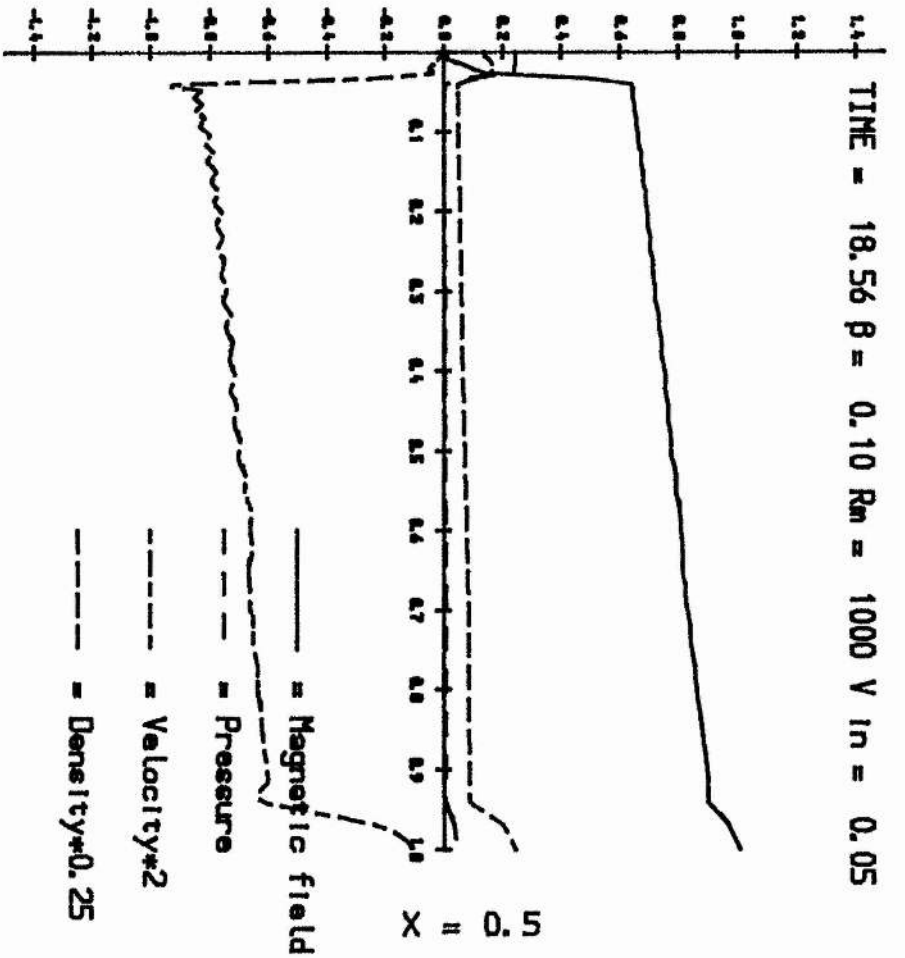


Figure 3.30 Magnetic field, pressure, velocity and density on the line x = 0.5; alpha = 0.3

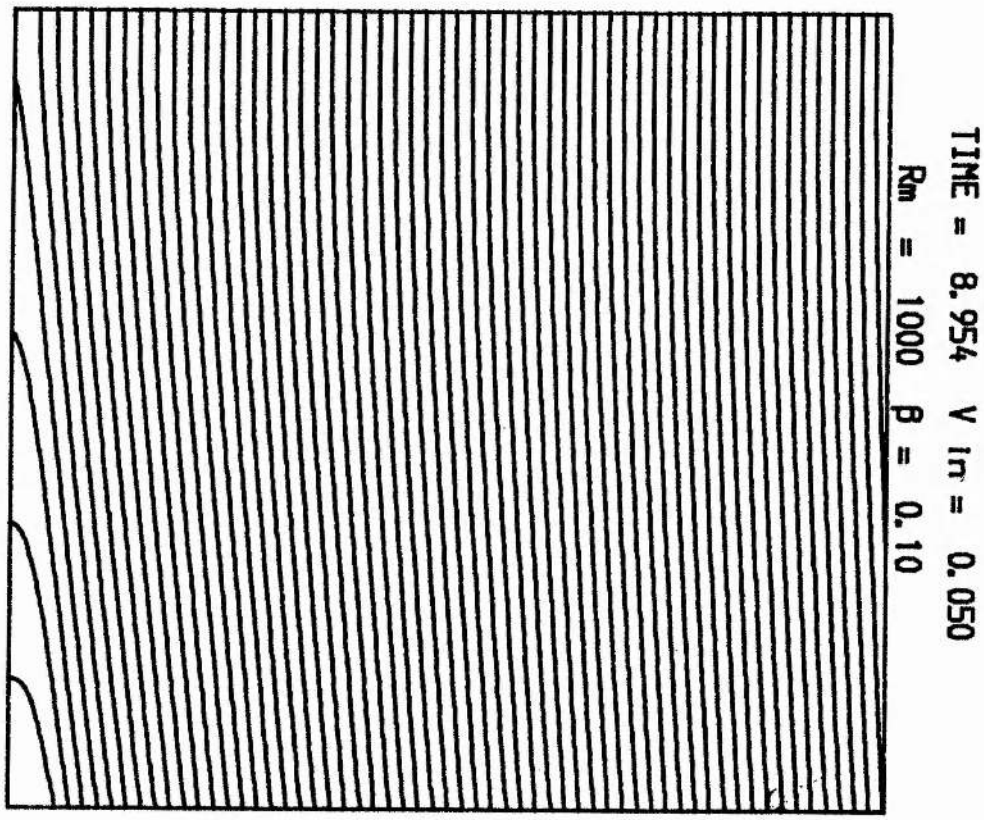


Figure 3.3I Contours of magnetic vector potential; alpha = 0.05

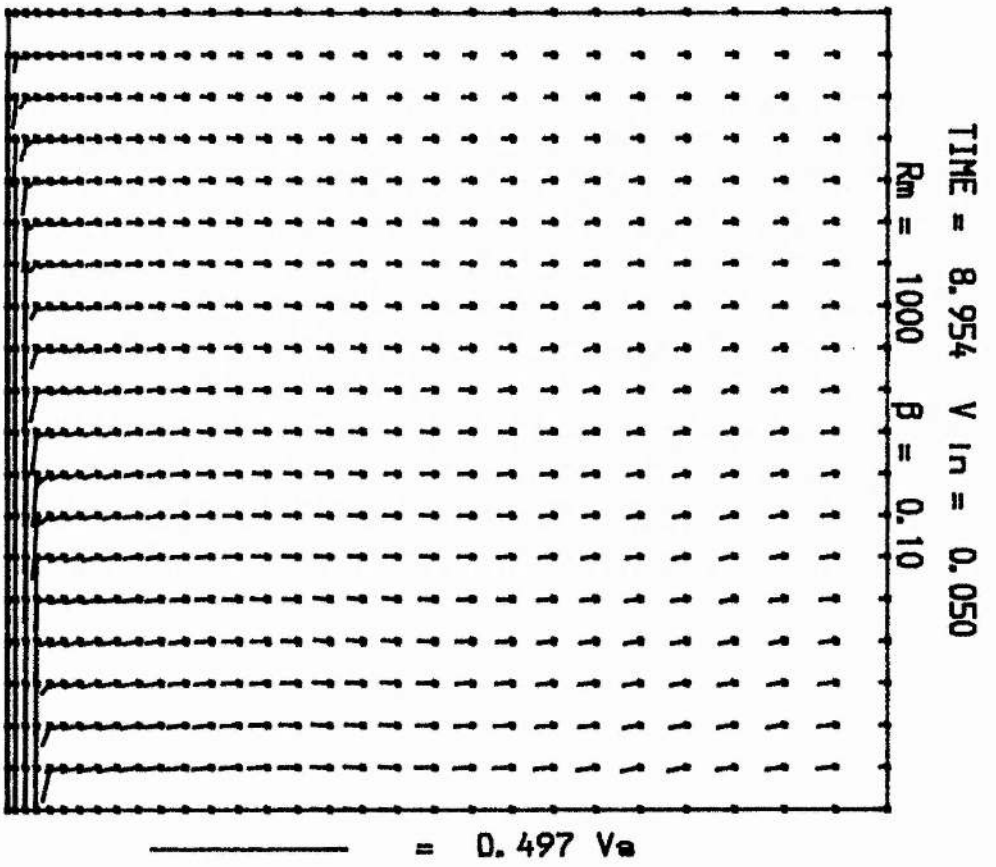


Figure 3.32 Plasma flow vectors; alpha = 0.05

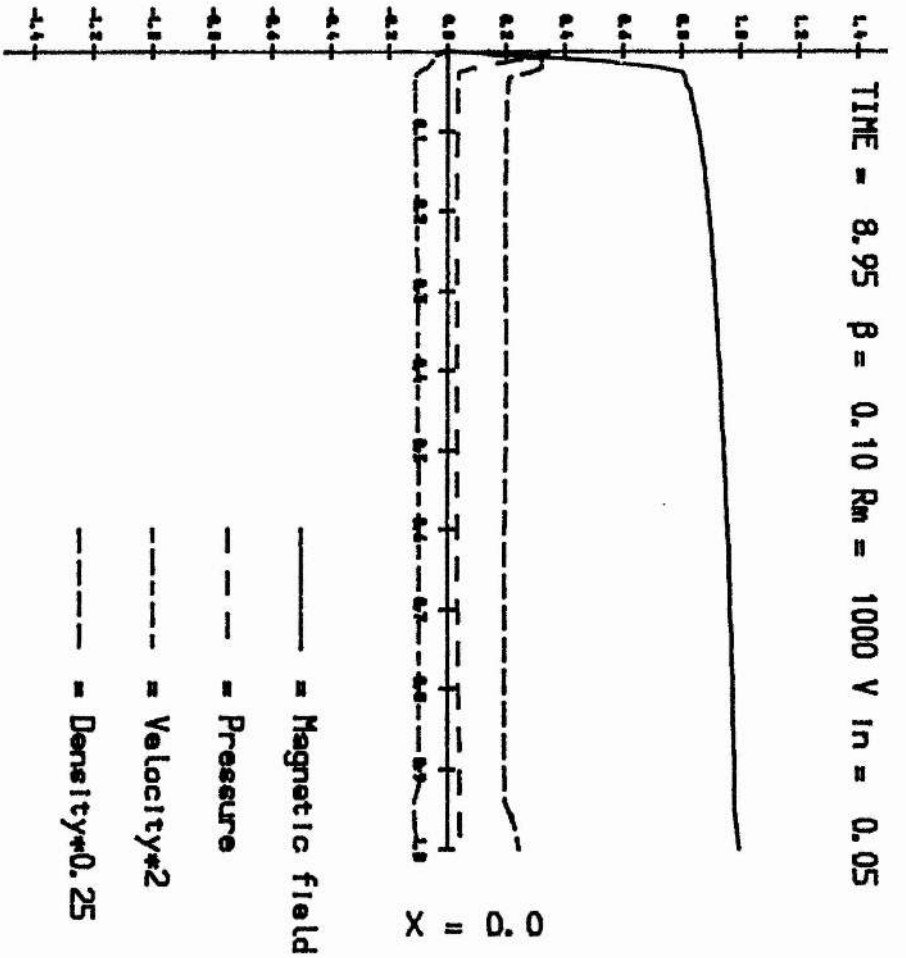


Figure 3.33 Magnetic field, pressure, velocity and density on the z-axis; alpha = 0.05

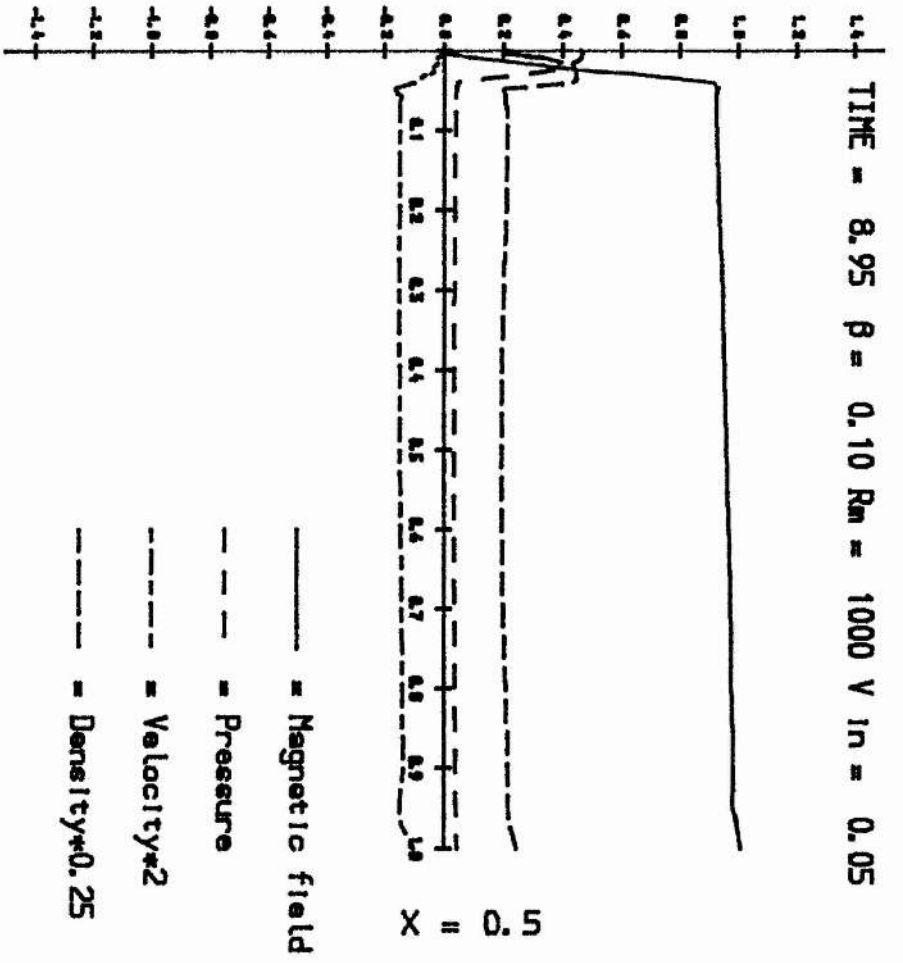


Figure 3.34 Magnetic field, pressure, velocity and density on the line x = 0.5; alpha = 0.05

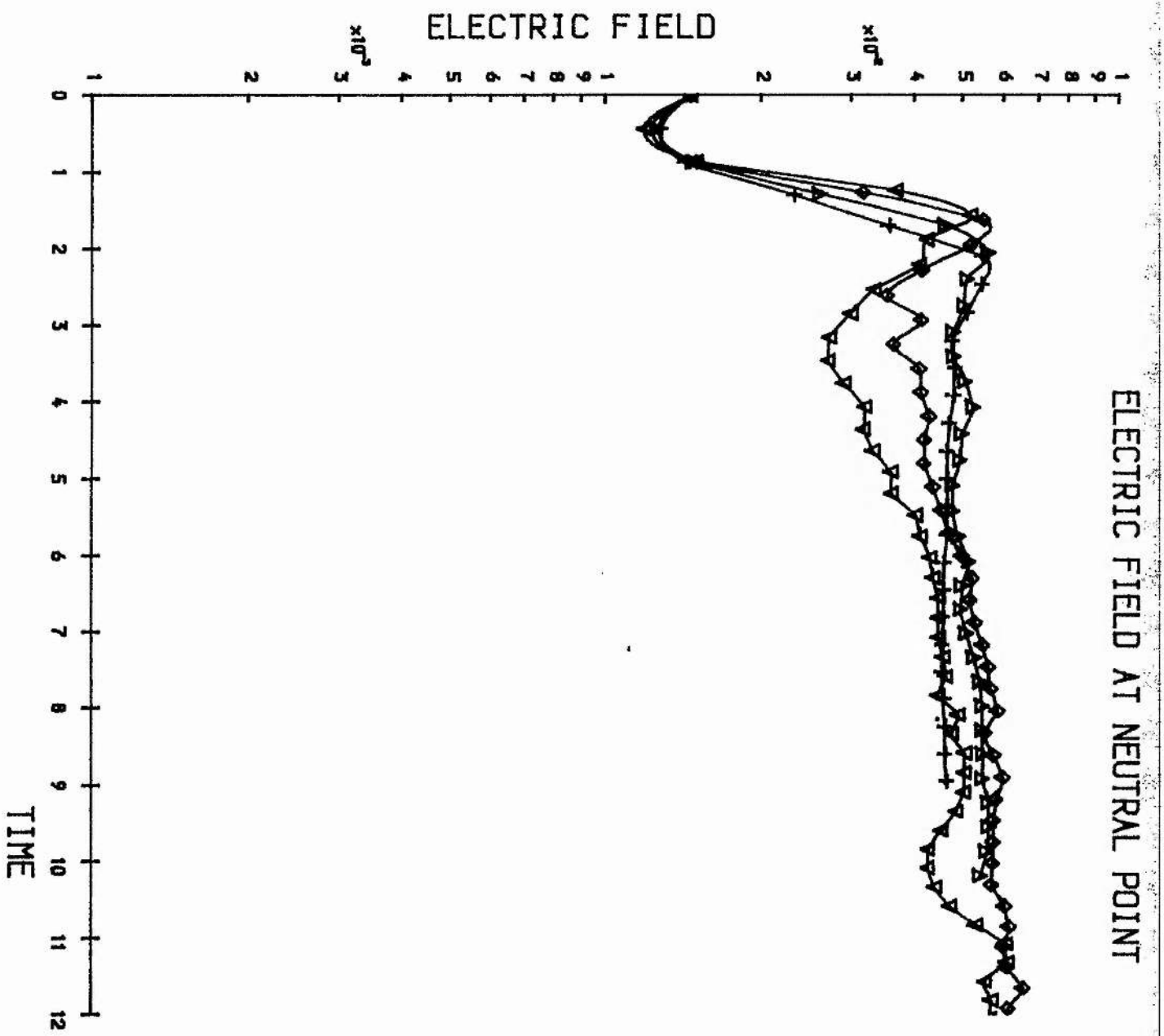


Figure 3.35 Rate of reconnection against time for different alpha

▽ : alpha = 0.05

△ : alpha = 0.20

◇ : alpha = 0.I0

+ : alpha = 0.30

that a numerical model which does not include curved incoming fields cannot simulate two-dimensional reconnection.

3.5.1 Size of the diffusion region

The edge of the diffusion region was defined as in the previous section. Results are presented in table 3.4.

TABLE 3.4 Rm = 1000 V in = 0.05

α	L	l	L/l
0.05	0.174	0.018	9.4
0.10	0.107	0.013	8.3
0.20	0.054	0.009	6.2
0.30	0.035	0.008	4.6

The more curved the field lines, the shorter and narrower the diffusion region becomes, as we might have expected from physical intuition. Again, we may examine the scaling of L and l as α varies (Figure 3.36). We find:

$$l \sim \alpha^{-1} \qquad L \sim \alpha^{-\frac{1}{2}}$$

which gives an aspect ratio

$$L/l \sim \alpha^{\frac{1}{2}}$$

We do not know of any other numerical studies in which this quantity has been estimated, so an independent check would be of especial interest.

3.5.2 Alfven Mach numbers

Petschek's theory assumes uniform, uncurved field lines at infinity. This is impossible to model within our numerical scheme, so a comparison between theory and experiment will be of limited validity. However, it

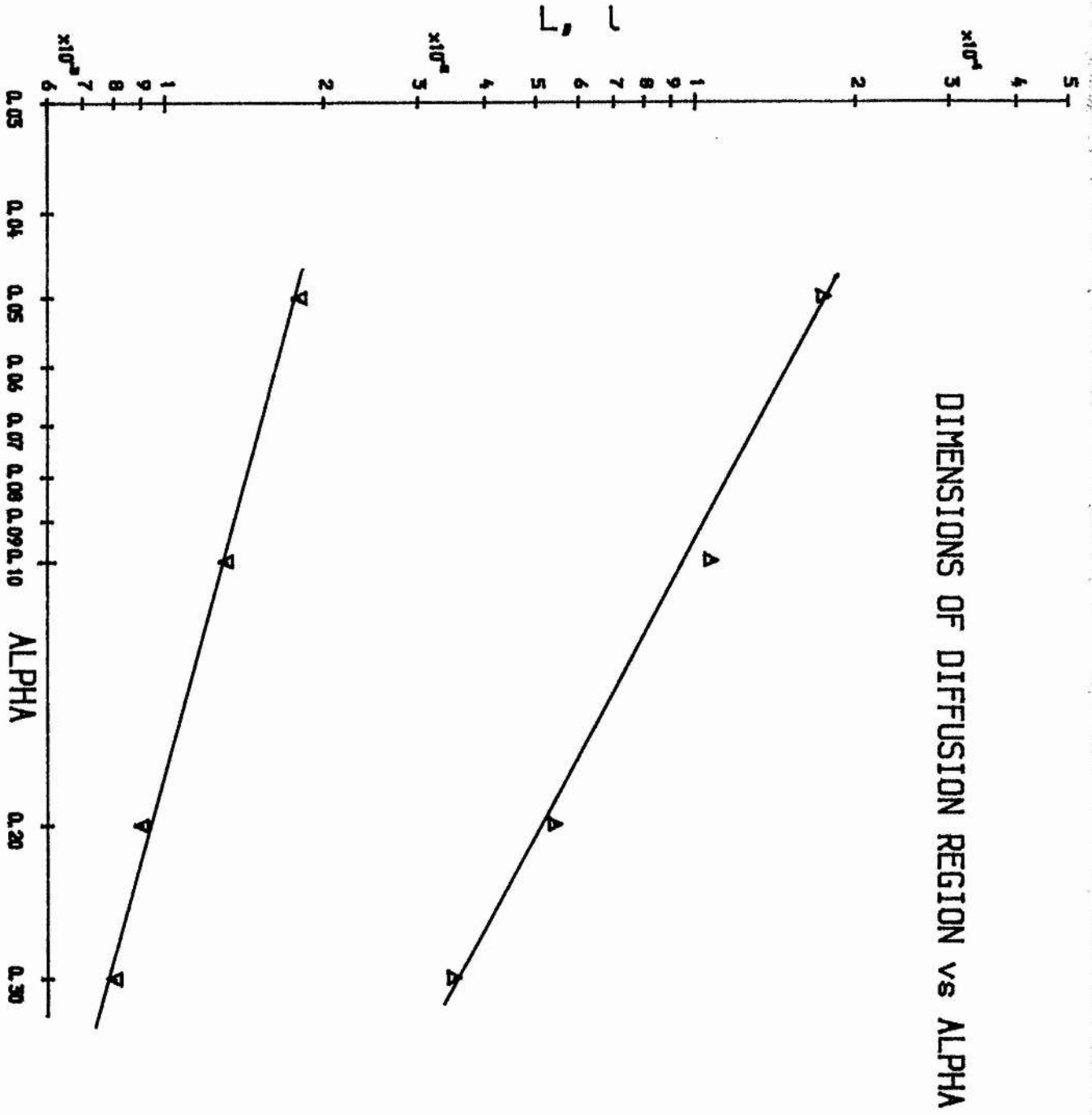


Figure 3.36 Dimensions of diffusion region against alpha

▽ : l (length of diffusion region in the z - direction)

△ : L (length of diffusion region in the x - direction)

is still of interest to examine the measured Alfven Mach numbers within the box.

TABLE 3.5 Rm = 1000 V in = 0.05

α	M_e	M'_e	M_i	E
0.05	0.043	0.065	0.055	0.046
0.10	0.055	0.099	0.072	0.054
0.20	0.062	0.152	0.082	0.058
0.30	0.042	0.215	0.086	0.058

M_e, M'_e, M_i are the Alfven Mach numbers measured with respect to the local Alfven speed at (0.0,0.8), (0.5,0.8), (0.0,0.1) respectively. The predicted Sweet-Parker reconnection rate is 0.032 and the predicted internal and external Petschek Mach numbers are 0.042, 0.057 respectively. The major differences apparent are the very high speeds appearing in the external region under high field curvatures, suggesting that current theoretical estimates for plasma velocities will severely underestimate the true value in island coalescence and similar experiments.

3.5.3 Variation of shock angle

θ here is measured in radians as in the previous section.

TABLE 3.6 Rm = 1000 V in = 0.05

α	θ
0.05	0.052
0.10	0.057
0.20	0.064
0.30	0.072

The shock angle has a weak dependence on field line curvature, a result suggested by Biskamp's simulation which has even higher α and a corresponding very high shock angle.

3.6 Variation of inflow speed

To investigate the effects of the boundary layer, and to try to decide whether its appearance was a physical or numerical artifact, we ran the code several times with widely differing inflow speeds. From an examination of the electric field and how the external Mach numbers varied with differing inflow speeds, we were able to deduce that the rate of reconnection is only weakly dependent on V_{in}.

TABLE 3.7 Alfven Mach numbers Rm = 1000

α	V_{in}	M_e	M'_e	M_i	E
0.05	0.05	0.043	0.065	0.043	0.046
0.10	0.05	0.055	0.099	0.072	0.054
0.10	0.50	0.076	0.168	0.078	0.050
0.05	0.001	0.053	0.077	0.075	0.055
0.30	0.001	0.044	0.206	0.089	0.053

For instance, consider the second and third runs. Although the inflow speed has increased by a factor of ten the internal Mach number has hardly changed at all, while M'_e has only increased by a factor of forty percent. The fourth and fifth runs show the outcome of a very low inflow rate; again, the system is reconnecting at its own rate rather than that imposed at the boundaries.

TABLE 3.8 Shock angles and size of the diffusion region

α	Me	Vin	θ	L	l	L/l
0.05	0.043	0.05	2.97	0.174	0.018	9.4
0.1	0.055	0.05	3.25	0.107	0.013	8.3
0.1	0.076	0.50	4.10	0.047	0.010	4.61
0.05	0.053	0.001	3.04	0.126	0.012	10.2
0.30	0.044	0.001	4.17	0.034	0.008	4.4

The first three runs shown indicate that the inflow speed, as measured by M_{∞}, appears to have a marked influence on the interior regions. With increased inflow speed the diffusion region shrinks, the aspect ratio lessens and the shock angle becomes wider.

The last two runs show the effect of a very low inflow speed, so that the system can effectively find the 'natural' reconnection rate. Here the field line curvatures are dominating the different inflow speeds, with a corresponding effect on the dimensions of the diffusion region.

3.7 Comparison with other published numerical work

A study presented by Dieter Biskamp (1985, 1986) of driven reconnection is closest in philosophy and approach to this work, and a comparison shows some interesting contrasts. Biskamp performs an incompressible calculation, solving the 2-D MHD equations with a finite-difference scheme of the predictor-corrector type over a numerical box with non-uniform grid spacing. In most cases his resolution is 128×512 points with non-uniform grid spacing, which enables him safely to take a higher magnetic Reynolds number than we could use.

The major difference between Biskamp's results and ours is his much longer current sheet caused by flux pile-up, the reverse current spike at the end of his current sheet and his very high shock angle. We believe that most of these artifacts can be explained by the *ad hoc* boundary conditions he assumes. The values of his field variables at the edges of the box are initially taken as given, and the system is advanced in time until the configuration is approximately steady. He finds that while there are no discontinuities at the inflow boundary a

strong boundary current layer (or shock) forms at the outflow region. This is eliminated by then assuming zero current in the region and by solving Poisson's equation to determine new boundary conditions. These are put back into the code which is advanced a few timesteps further to allow all variables to readjust. Repeating the procedure several times effectively eliminates the boundary layer.

We suggest that this procedure is not allowing the outflow boundary to act as a true 'free-floating' region. No such current layers have been observed at the corresponding boundary in our simulations, and it seems likely that Biskamp's algorithm for treating the edge is inhibiting the exit of plasma and flux, causing a 'pile-up' at the boundary. Magnetic field enters the box but cannot be convected towards the current sheet at the normal rate because of the numerically imposed 'choke' at the exit boundary, which leads to the observed accumulation of flux. Also, the shock angle is much greater than it should be because of the resulting magnetic pressure beyond the separatrix.

3.8 Conclusions

We have presented a series of numerical solutions to the steady-state two-dimensional MHD equations. These have shown many features common to the classical Petschek picture, but also several new properties, especially when the inflow parameters are varied. Of these, the most noticeable are the behaviour of the plasma flow vectors (indicating an interplay between the flow pattern imposed from outside

and naturally arising flows from magnetic reconnection) and the detailed structure that appears within the current sheet.

We have varied three of the inflow parameters to see how the system responds. The behaviour is here summarised:

(i) Variation of plasma conductivity has a strong effect on the reconnection rate, the dimensions of the diffusion region and the shock angle. The length of the diffusion region scales as the reciprocal of the magnetic Reynolds number. Flux pile-up and resultant magnetic tearing is observed for sufficiently high Rm. The scaling of the measured Mach numbers suggests that Petschek's theory is holding in the internal region, while the inflow parameters (field line curvature, inflow speed) dominate the plasma speeds in the outer region.

(ii) Varying the field line curvature does not significantly alter the rate of reconnection, but does strongly alter the global flow pattern of the plasma. The length of the diffusion region scales inversely as the field line curvature, and there is a weak dependence of the shock angle on this parameter.

(iii) Varying the inflow speed does affect the internal region. Increasing this paramater widens the shock angle and decreases the dimensions of the diffusion region.

All solutions (except for the one showing unsteady flux pile-up) have shown properties characteristic of a weak fast-mode expansion; magnetic field and plasma pressure decrease on approaching the current sheet. The only parameter to control the rate of reconnection is the magnetic conductivity, and the rate of reconnection is of order $1/\log(Rm)$.

In the absence of curvature on the incoming field lines the plasma flow vectors will show convergence as they approach the current sheet. However, with even a small curvature the vectors diverge, as in stagnation-point flow. We expect that this will be characteristic of most physically observed flows.

Our study was unable to vary Vin over a sufficiently wide range to say how significant an effect this parameter will have on the system. From the measurements made, it seems that the reconnection rates will not be affected.

These simulations have shown that a complex interaction exists between the internal and external conditions on a reconnecting MHD system. While the rate of the reconnection is governed by the plasma resistivity in all cases, the external flow patterns will dominate the flows naturally arising in the limit of an isolated system. The dynamics will be further complicated by non-linear energy cascades from low to high wavenumbers, causing plasma turbulence on the lowest length scales, and leading to current filamentation, microturbulence and anomalous resistivity (see Chapter 1, Mattheus and Montgomery (1981), Stenzel et al (1985).) Further, it is not always possible to self-consistently impose an arbitrary set of boundary conditions, as certain restrictions arising from both the physics and the mathematical consistency of the system must always be satisfied.

More computational experiments at high resolution are required to understand the nature of these flows and the detailed interaction between all length scales, but this will require an increase in computing power of two or more orders of magnitude than we had available. Also important will be development of boundary conditions

using the full compatability equations, which we have found to be at least as important as the numerical scheme used within the code. Clearly, there is a need for much further work on this problem.

CHAPTER 4

TEARING MODES

4.1 Introduction

Chapter 4 describes how our code was adapted to model tearing modes in Cartesian geometry with two different sets of boundary conditions. For the general analytical theory of tearing modes, the reader is referred to the original paper by Furth, Killeen and Rosenbluth (1963), together with the review papers by Wesson (1982), Priest (1985) and Furth (1985).

For both simulations our initial configuration was an anti-parallel magnetic field with resulting current sheet, The field was allowed to tear so that one magnetic island formed. The upper boundary (parallel to the initial current sheet) was freefloating, and thus allowed plasma and flux freely to enter and exit the box. The difference between the two sets of simulations was the nature of the boundary on which the O-point lay. For our first set of runs, we chose to make this edge an axis of symmetry, so that with a suitable initial perturbation the program was then intended to give results to be compared with the well-developed analytical theory. Although identical in all other respects, the second set of runs had a freefloating boundary at this position, leading to a configuration much closer to that studied by theorists of driven reconnection.

One of the first simulations in this area was performed by Wesson (1966), who solved the incompressible MHD equations to confirm the growth rates predicted by Furth, Killen and Rosenbluth (1963). Since then, much has been published on the dynamics of the tearing mode because of its importance in the context of magnetic confinement of

tokamak plasmas. Firstly, the change in field topology allowed by magnetic tearing can enhance transport of impurities across the plasma, which then spoils the containment properties. Secondly, various manifestations of the tearing mode are believed to lead to Mirnov oscillations and the disruptive instability, which causes a collapse of the plasma current.

In particular, the last decade has seen a proliferation of studies concerning numerical simulations of the tearing mode, both in cylindrical and toroidal geometries for application to laboratory plasmas, and in Cartesian geometry; here we are more concerned with the latter class. We describe three of the most relevant experiments:

Birn (1980) reports on a numerical experiment in Cartesian geometry, for which the initial configuration is a current sheet with a transverse field component included. The length of his box was roughly the wavelength of the fastest growing mode, and the boundary conditions were freefloating. With a magnetic Reynolds number of 500 he finds that initially a pair of X and O-points form, after which a steady state appears.

Steinolfson and Van Hoven (1983) have studied the linear development of tearing in a current sheet with rather higher magnetic Reynolds numbers (up to 10^{12}) and dimensionless wavelengths (up to 10^{3}) than the values normally attainable. They assume uniform resistivity within an incompressible, inviscid plasma, and infinite-medium boundary conditions at large x. Among their main findings are that the constant-ψ approximation holds for large (e.g. $k = 0.5$) wavenumbers, and that the growth rate scales as $S^{2/5}$. For longer wavelengths B_{1x} varies across

the diffusion layer , so the constant- ψ approximation is no longer valid, and the growth rate scales instead as $S^{2/3}$.

Forbes and Priest (1982) model a two-dimensional configuration of a current sheet line-tied at one end, which is then allowed to tear spontaneously. They use the same flux-corrected code as we do in the study presented below, and find growth rates and a tearing region width roughly consistent with the analytical theory. The calculation is allowed to proceed until around 130 Alfven transit times, when the calculation beomes numerically unstable. By the end of the calculation a quasi-steady state has been reached. The velocity of the inflowing plasma at the current sheet has a value consistent with the nonlinear reconnection theory of Soward and Priest (1977).

Section 4.2 describes the implementation of the symmetric tearing mode problem, and describes fully the boundary conditions and initial perturbations used. These comprise the only essential difference between the code used here and that in the previous chapter, so we omit a detailed description of the algorithms employed and refer the reader to Chapter 2 for an account of the techniques used. In section 4.3 we give a similar account of the open-ended tearing problem. Section 4.4 compares the experimental results from various runs of the code with different initial perturbation levels and magnetic Reynolds numbers. Growth rates for the mode as measured by the reconnection rate at the resulting X-point are shown as functions of time. We conclude with some comparisons and comments in Section 4.5.

Section 4.2 Numerical implementation of the symmetric tearing mode

Figure 3.4 shows the geometrical configuration we used to simulate tearing modes. The same labelling for the walls of the numerical box is used as in Chapter 3, as are the equations, non-dimensionalisation of the variables, geometry and non-uniform gridspacing.

The first configuration we chose to study had three axes of symmetry enclosing the current sheet and one freefloating boundary:

Surface 1 (symmetry)

$$V_x = B_z = \frac{\partial B_x}{\partial x} = \frac{\partial v_z}{\partial x} = \frac{\partial \rho}{\partial x} = \frac{\partial p}{\partial x} = 0 \qquad (4.2.1)$$

Surface 2 (symmetry)

$$V_z = B_x = \frac{\partial B_z}{\partial z} = \frac{\partial v_x}{\partial z} = \frac{\partial \rho}{\partial z} = \frac{\partial p}{\partial z} = 0 \qquad (4.2.2)$$

Surface 3 (symmetry)

$$V_x = B_z = \frac{\partial B_x}{\partial x} = \frac{\partial v_z}{\partial x} = \frac{\partial \rho}{\partial x} = \frac{\partial p}{\partial x} = 0 \qquad (4.2.3)$$

Surface 4 (freefloating)

$$\frac{\partial B_x}{\partial z} = \frac{\partial v_x}{\partial z} = \frac{\partial v_z}{\partial z} = \frac{\partial \rho}{\partial z} = \frac{\partial p}{\partial z} = 0 \qquad \frac{\partial B_z}{\partial z} = -\frac{\partial B_x}{\partial x} \qquad (4.2.4)$$

Imposing two parallel axes of symmetry at either end of the current sheet constrains the tearing mode to have a maximum wavelength of 0.25 in our dimensionless units. We found in all simulations with these boundary conditions that an X-point always formed at one corner and an

O-point at the other, suggesting that the 'natural' or fastest growing wavelength for this configuration has a quarter-wavelength greater than the length of the box. One experiment we would have tried, given more time, would have been to increase the dimensions of the computational domain and to trigger several wavelengths of the tearing mode instability.

Because of the different nature of this problem from that treated in the previous chapter, we also had to rewrite the vector potential conversion routine. Use was made of the freefloating boundary by integrating the x-component of B outwards from the current sheet to the freefloating edge where A was not specified, but calculated by extrapolation from its value within the box, and thus we avoided the 'overspecification' problem for two boundary conditions on a solution to a first order system.

Successful implementation of an initial perturbation that would start magnetic tearing proved to be the hardest problem in this study. In some studies a perturbation is not necessary to start reconnection. Forbes and Priest (1982) performed a numerical experiment that simulated the main phase of a two-ribbon solar flare; here an asymmetry was already present in the line-tied boundary conditions, which eventually triggers reconnection because of truncation error in the computation and resulting low-level numerical noise. Conversely, we found that the high degree of symmetry in our problem and the relatively low magnetic Reynolds numbers meant that much higher levels of perturbation were required to start up tearing, since Ohmic diffusion played a dominant role in the initial development of the current sheet and could damp out a low-level pertubation. The minimum level of perturbation necessary to

start tearing was thus a function of the magnetic Reynolds number. These perturbations are decsribed more fully in Section 4.3. In particular, we found that attempting to trigger tearing from the numerical noise inherent in such a calculation did not work. It seems that Forbes and Priest found this approach to be successful because their line-tied boundary conditions had an inhibitory effect on Ohmic diffusion.

Some experimentation was also necessary to find a workable mathematical form for the perturbation and the best variable to perturb. Firstly, we tried localising the initial perturbation about the x-axis according to the prescription

$$Bz = \alpha \exp(-20z) \sin(\pi x/2) \qquad (4.2.5)$$

where α was varied down to 0.01. This led to tearing-type patterns in the flow vectors and the formation of one quadrant of a magnetic island. Unfortunately, the code went unstable after several Alfven scale times for any value of the magnetic Reynolds number over 2500, which severely curtailed the planned course of study.

This behaviour may be interpreted in the light of Arter and Eastwood's theory (1986). This first perturbation generated harmonics at high wavenumbers which led to an energy cascade through to the lowest wavelengths and through the 'aliasing' instability to the other end of the Fourier spectrum, causing a positive feedback instability. Chapter 2 gives a fuller account of this phenomenon.

A different, workable perturbation was eventually arrived at after more experimentation. We chose to add a small random component to one of the independent variables, as did Steinolfson (1981) and Schnack (1978).

Trying their approach of perturbing the velocity field did not work for any of the Reynolds numbers we could take, and it seems that with our low plasma β (of 0.1) the initial anti-parallel magnetic field was unaffected by hydrodynamic pertubations, or at least dominated by one-dimensional diffusion.

With this in mind we chose to add a random component onto the magnetic field. To conserve the zero-divergence condition we added a perturbation onto B_z, called the subroutine AFROMB to derive the magnetic vector potential, and then called BFROMA to calculate B_x. The maximum level of the perturbation is denoted below by α. With sufficiently high α we found that this did start up a tearing flow. The code was found to run indefinitely, suggesting that either this perturbation did not contain the high amplitude, low wavelength harmonics of the first perturbation, or that these harmonics were damped out by magnetic diffusivity.

Section 4.3 Implementation of the open-ended tearing problem

To give results that may be compared to those from the previous section, we implemented a different set of boundary conditions. In this case there are two adjacent axes of symmetry and two adjacent free-floating boundaries.

Surface 1 (symmetry)

$$v_x = B_z = \frac{\partial B_x}{\partial x} = \frac{\partial v_z}{\partial x} = \frac{\partial p}{\partial x} = \frac{\partial \rho}{\partial x} = 0 \qquad (4.3.1)$$

Surface 2 (symmetry)

$$v_z = B_x = \frac{\partial B_z}{\partial z} = \frac{\partial v_x}{\partial z} = \frac{\partial \rho}{\partial z} = \frac{\partial p}{\partial z} = 0 \qquad (4.3.2)$$

Surface 3 (freefloating)

$$\frac{\partial B_z}{\partial x} = \frac{\partial v_x}{\partial x} = \frac{\partial v_z}{\partial x} = \frac{\partial \rho}{\partial x} = \frac{\partial p}{\partial x} = 0 \quad \frac{\partial B_x}{\partial x} = -\frac{\partial B_z}{\partial z} \qquad (4.3.3)$$

Surface 4 (freefloating)

$$\frac{\partial B_x}{\partial z} = \frac{\partial v_x}{\partial z} = \frac{\partial v_z}{\partial z} = \frac{\partial \rho}{\partial z} = \frac{\partial p}{\partial z} = 0 \quad \frac{\partial B_z}{\partial z} = -\frac{\partial B_x}{\partial x} \qquad (4.3.4)$$

This configuration may be compared to corresponding simulations from the previous chapter; here the difference is that plasma and flux may flow over the inflow boundary at a rate determined by the internal behaviour of the plasma, rather than by externally determined constraints. With this in mind we biased the initial perturbation so as to initiate an x-point at the lower left-hand edge of the box (at the corner between the axes of symmetry). Without this, we found that an O-point formed preferentially and that a large island formed at the centre of the box about which plasma flowed before leaving the box. X-point flow was generated by imposing a positive, random perturbation on B_z, which produced good results.

Section 4.4 Experimental results

In this section we present some results from two 'typical' runs under our two different sets of boundary conditions (Figures

4.2-4.13, 4,15-26). Plots of magnetic field and flow vectors are given at several different times, since tearing is a highly time-dependent process.

Tables 4.1 and 4.2 give details of the two sets of experiments:

Table 4.1 'Closed' reconnection

(Three axes of symmetry, one freefloating axis)

Rm	α
1000	0.01
2000	0.01
1000	0.05
5000	0.05
10000	0.05

Table 4.2 'Open-ended' reconnection

(Two axes of symmetry, two freefloating axes)

Rm	α
1000	0.01
10000	0.01

An examination of Figure 4.1 shows directly the first major difficulty we encountered with these simulations. The graphs plot the electric field at the X-point and O-point as a function of time for three different runs. Analytic theory of the tearing mode instability predicts that in the linear phase (at, say, less than one tearing time) the electric field at the X-point will rise exponentially . In fact, our experiment for Rm = 1000 shows that the field decreases in time. We interpret this as a consequence of the low magnetic Reynolds number. The electric field at the current sheet will be affected by both the tearing-generated field and the initial field, which will be subject to

Ohmic diffusion. With a high diffusivity the current sheet will decay and the consequent decrease in electric field will swamp out any increase from the tearing. Only when higher magnetic Reynolds numbers are taken do the two effects start to counter each other (Rm = 5000) or the tearing to dominate the diffusion (Rm = 10000). It therefore appears that the results we present must have Ohmic diffusion taken into account in the analytical theory before a proper comparison can be made with other published work. For this reason we do not attempt to compare our growth rates with the analytical theory of Furth, Killen and Rosenbluth (1963).

The second difficulty was related to the first, in that it also involves the effects of diffusion on the tearing mode. We found that even when a suitable initial perturbation had been found, there existed a lower limit on its magnitude below which tearing would not start. This behaviour was also related to the initial diffusion of the current sheet, as it appears that when the level of perturbation is too low the tension in the magnetic field lines eliminates the imposed component of magnetic field perpendicular to the zero-order field, which is required to initiate tearing. However, even with the highest magnetic Reynolds numbers used in these simulations the perturbation level had to be so great that the tearing mode passed directly through the linear growth phase into the algebraic growth and saturation regimes. This is the reason for the very rapid appearance of the magnetic island compared to the tearing time in all simulations. However, the effect is probably unimportant in the second set of simulations where a steady state is set up (see below).

The first set of simulation results we present in detail shows 'closed' reconnection (three symmetry axes, one freefloating axis on the boundaries). A magnetic Reynolds number of 10000 was taken and an initial maximum perturbation level of 0.05 of the external ambient magnetic field used. The time development of the magnetic field and plasma flow pattern are shown in Figures 4.2 - 4.13. The plots of flow vectors show that at the early stages a standard 'tearing-type' flow is set up with an inflow at the X-point, fluid transport along the length of the island and outflow at the O-point. At later times the flows slow down due to dissipative effects, implying that the tearing mode is nearing saturation (note that the computer plots presented here are scaled so that the flow vector with the highest magnitude has the same length for all pictures).

An examination of the plots for the second case shows the major difference between the behaviour of the system under a different set of boundary conditions: the plasma jet leaving the computational domain for the 'open-ended' case. We present the results from a run with two axes of symmetry and two freefloating axes on the boundaries of the box, a magnetic Reynolds number of 1000 and an initial perturbation level of 0.01 (Figure 4.15 - 4.26).

The graphs of electric field (Figure 4.14) and plasma flow patterns suggest that, in contrast to the previous experiment, a steady reconnecting state has been set up. Plasma and magnetic flux are slowly convected into the box and are then expelled at high velocity when reconnection has occurred. Further, the graphs of electric field for two runs with these boundary conditions under different magnetic Reynolds numbers show different reconnection rates. This suggests that our

conclusion from Chapter 3, stating that reconnection rate is strongly dependent on Rm, is being confirmed.

Section 4.5 Conclusions

We have succesfully used the computer code SHASTA to generate a magnetic field configuration and the associated flow patterns that arise when a tearing mode is initiated in Cartesian geometry. Despite problems encountered with restrictions to low magnetic Reynolds numbers, we have shown that the nature of the tearing flow may have significant differences according to the boundary conditions imposed in the perpendicular coordinate to the current sheet. We have found that if both such edges are symmetry boundaries, tearing-type flows appear but naturally are constrained to within the region of simulation. By contrast, allowing plasma to freely enter and exit the box across this boundary results in the formation of two plasma jets within the current sheet, one entering and one leaving the box. We note that the relatively crudely numerically implemented free-floating boundary did not seem to affect the plasma behaviour, perhaps because the system entered a steady state soon after the experiment was started and so transient waves soon damped out or left the box. We were not so concerned with the apparent fastest-growing wavelength in the 'open-ended' experiments to exceed the length of the box, as we still have an experiment which may be directly compared with the findings from the previous chapter. In particular, it seems that most of the flow patterns observed in Chapter 3 were an artifact of the externally imposed magnetic field on the boundary; without this, the results from Chapter 4 show that tearing flow patterns

appear instead. The differences are highlighted in the conclusions in Chapter 6 of this thesis.

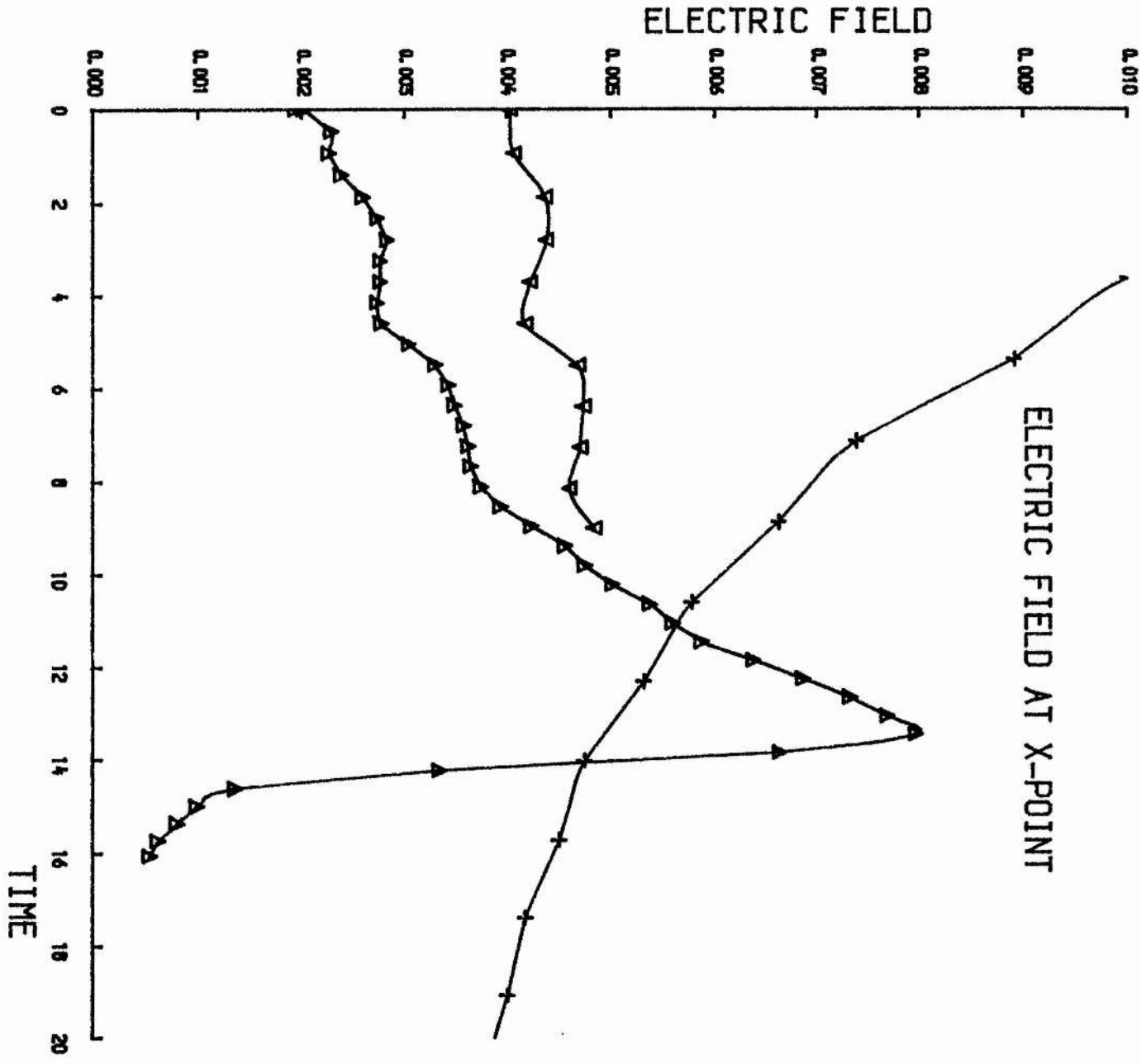


Figure 4.I Electric field at X-point for different R_m

+ : R_m = 1000

∇ : R_m = 5000 Initial perturbation level = 0.05

△ : R_m = 10000

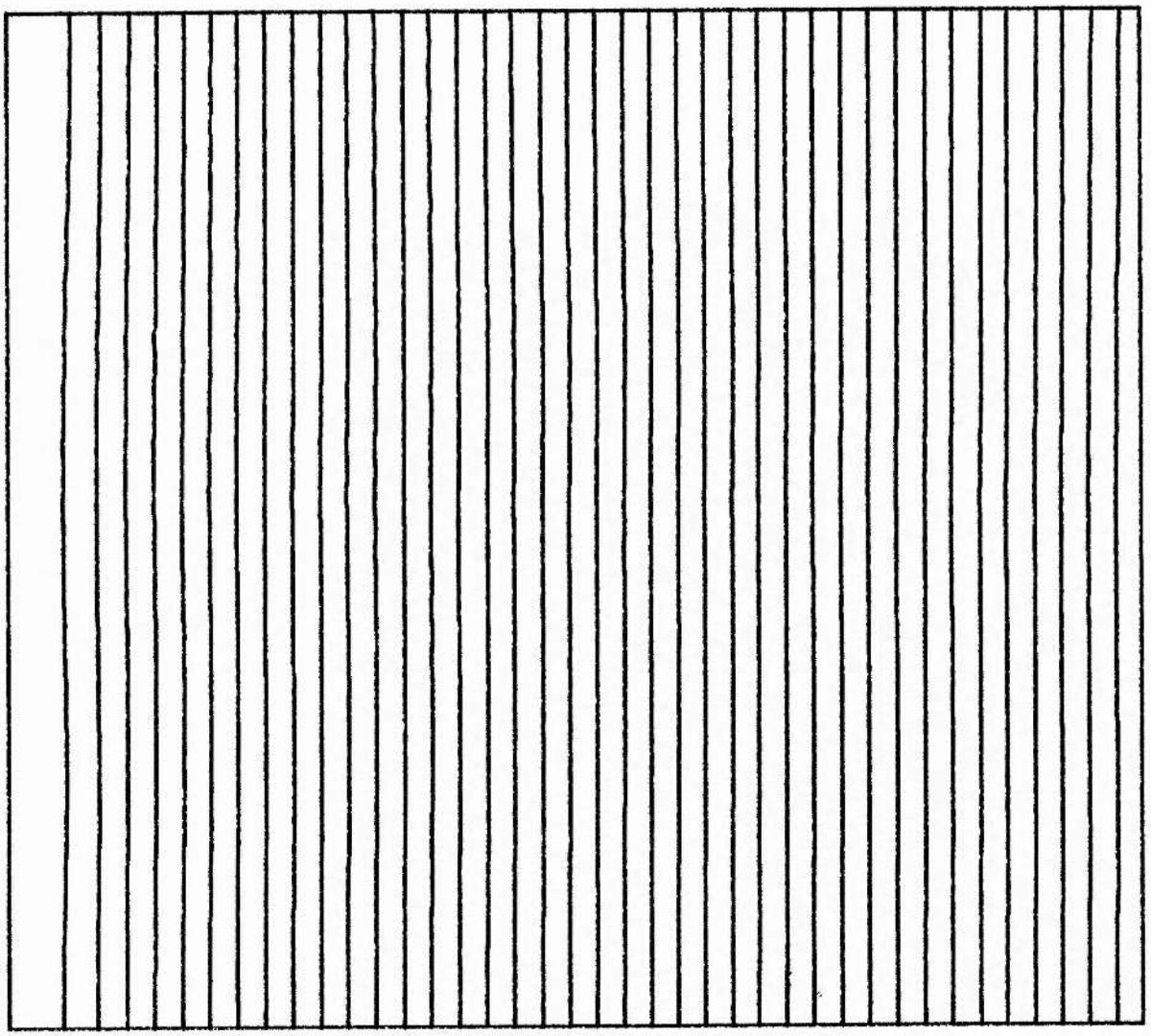


Figure 4.2 Magnetic field lines

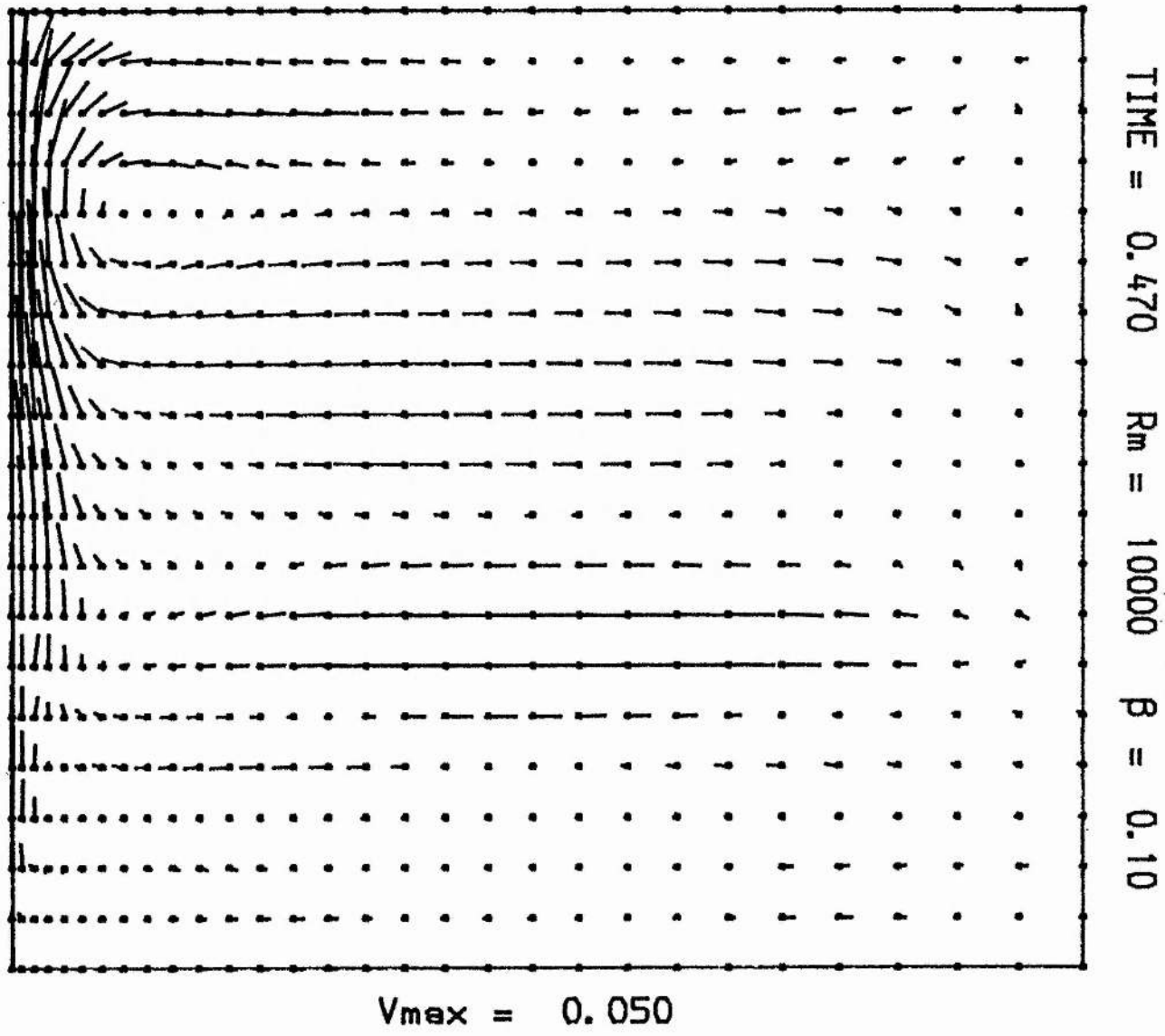


Figure 4.3 Plasma flow vectors

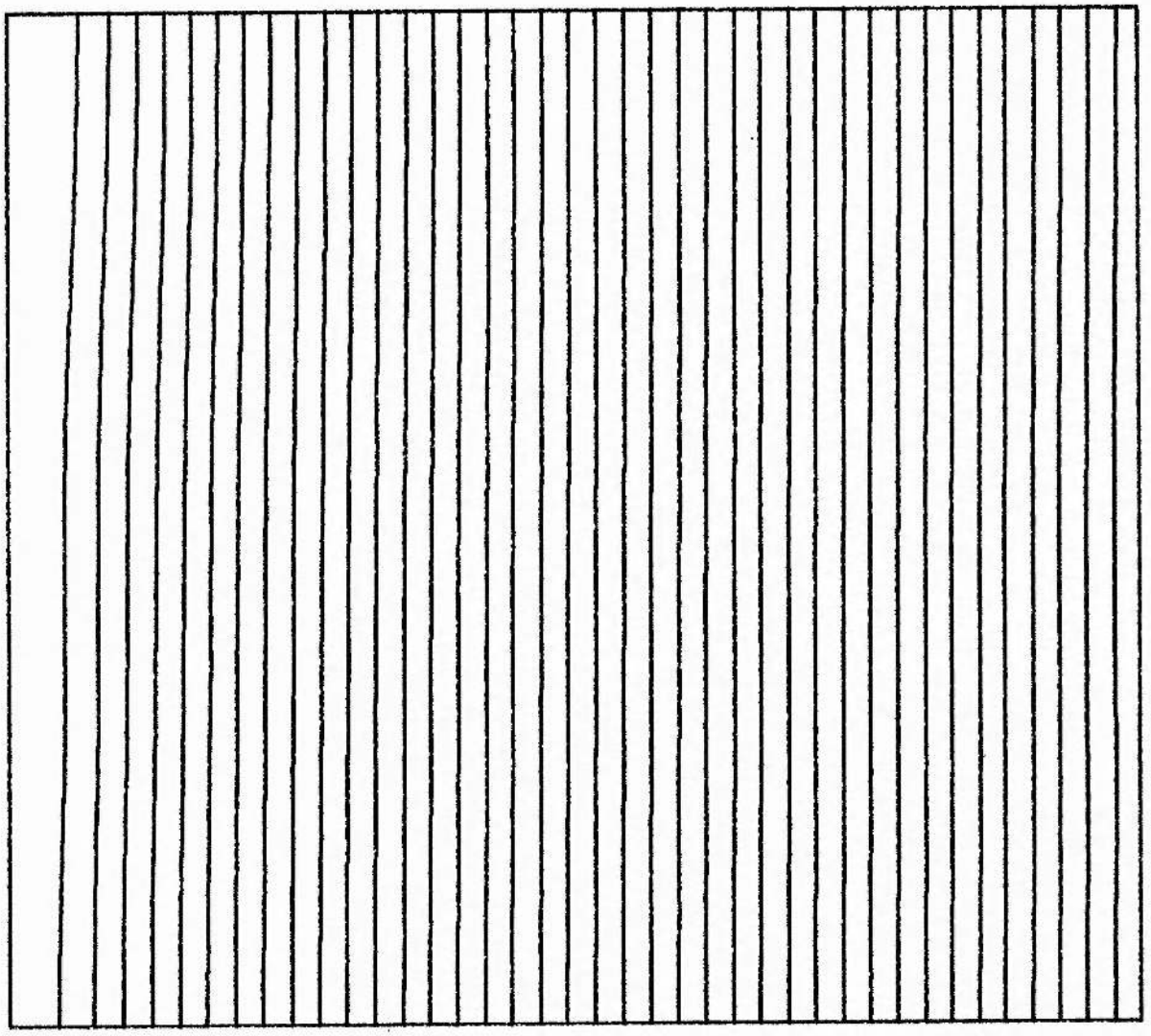


Figure 4.4 Magnetic field lines

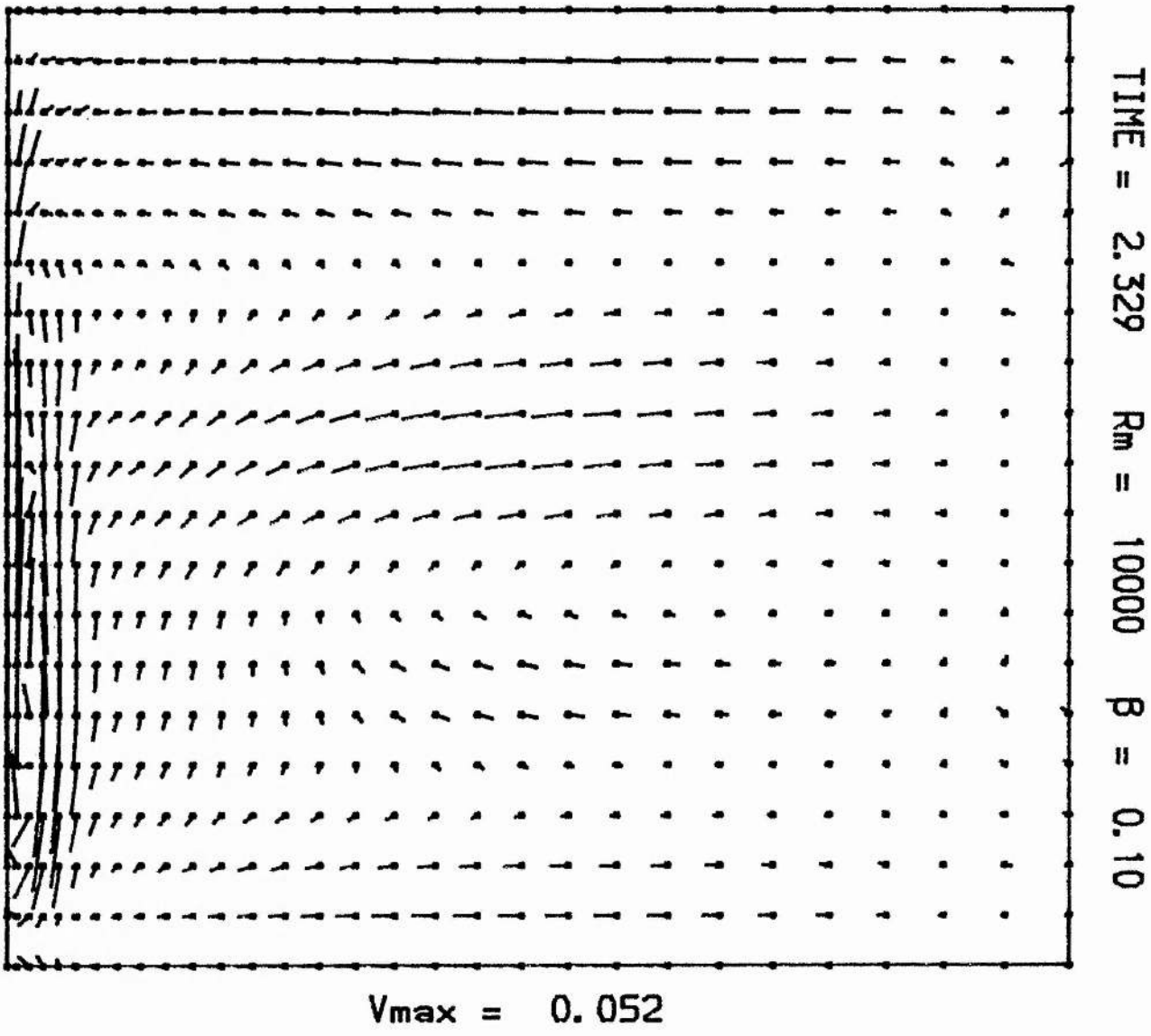


Figure 4.5 Plasma flow vectors

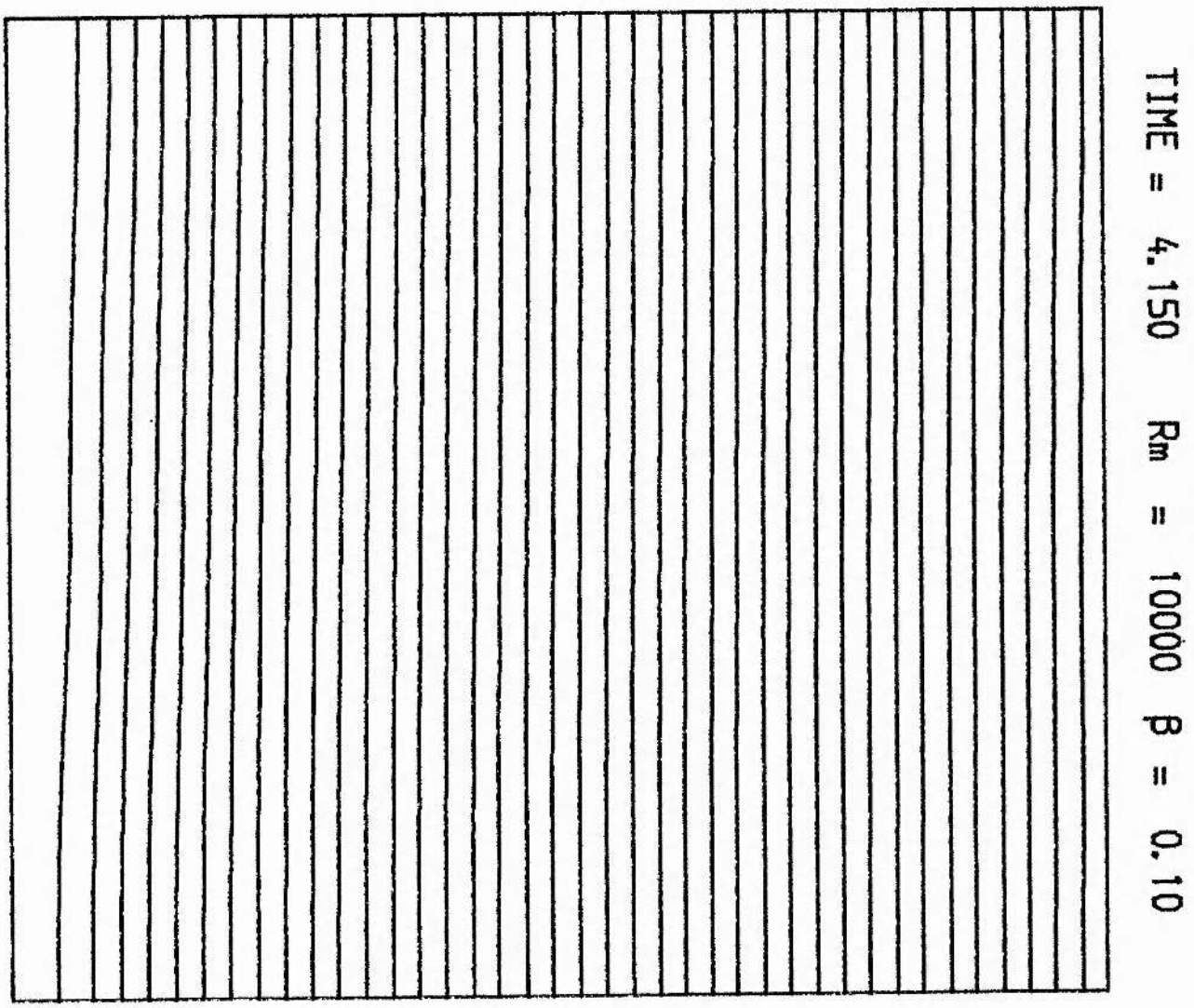


Figure 4.6 Magnetic field lines

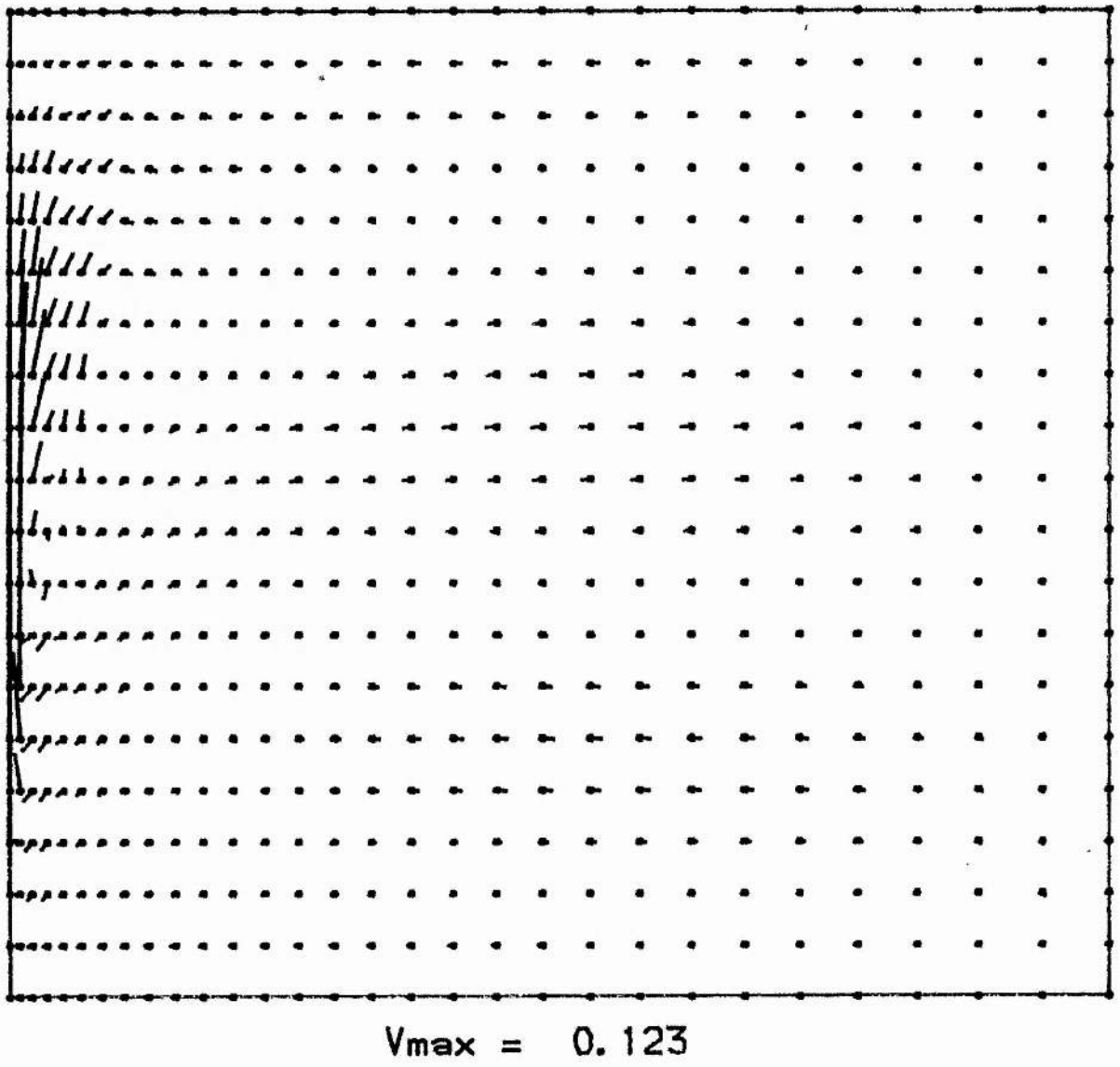


Figure 4.7 Plasma flow vectors

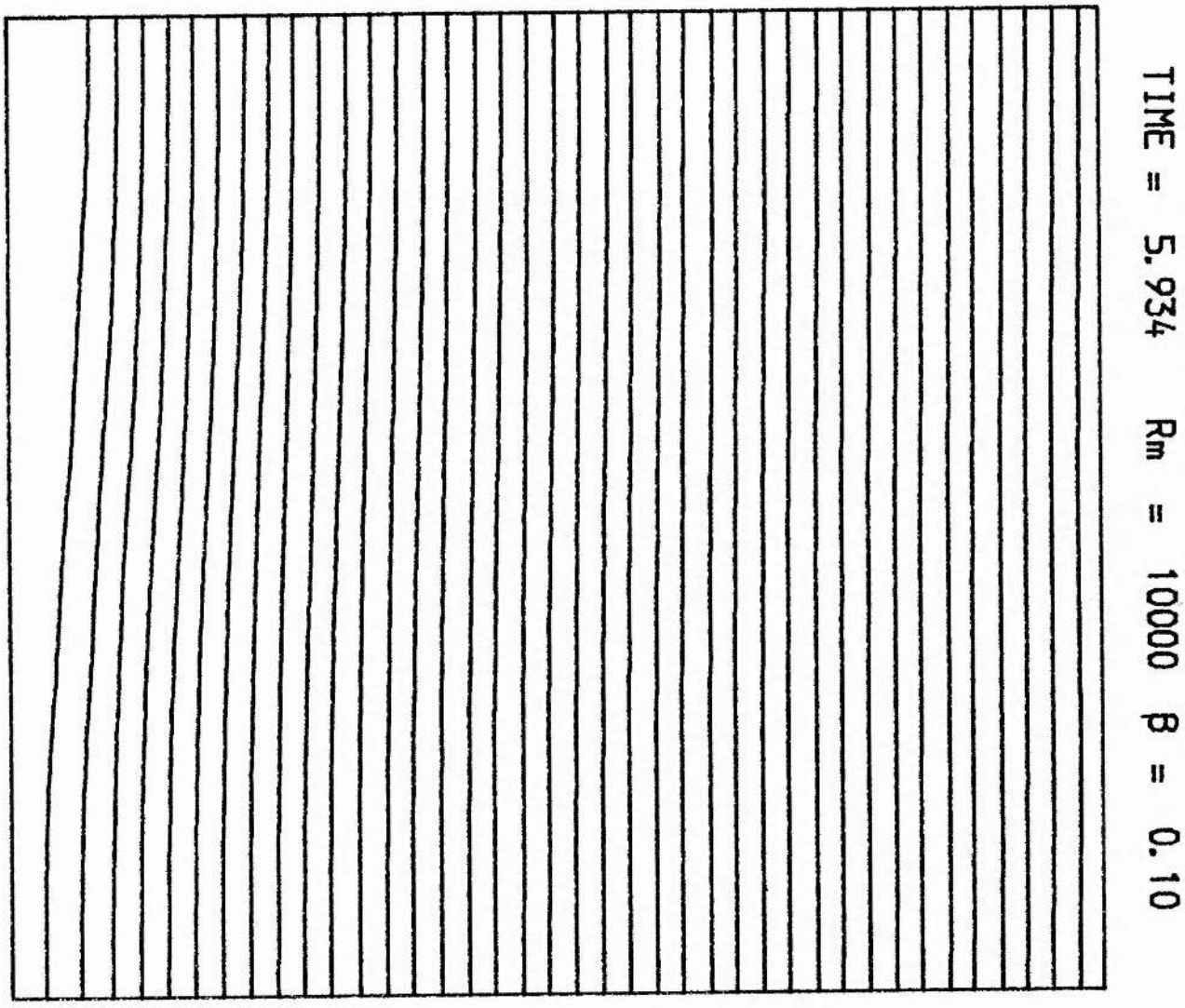


Figure 4.8 Magnetic field lines

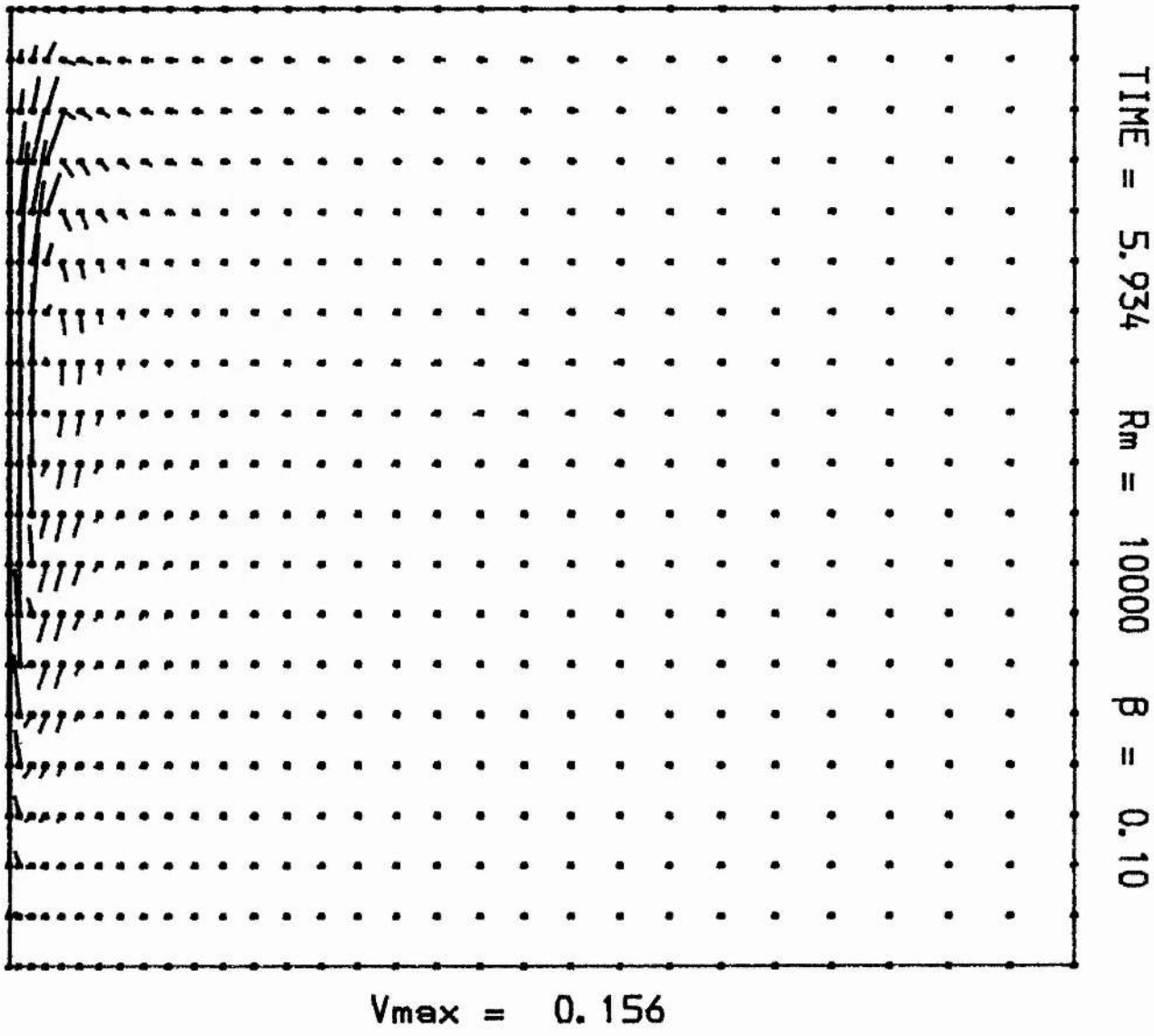


Figure 4.9 Plasma flow vectors

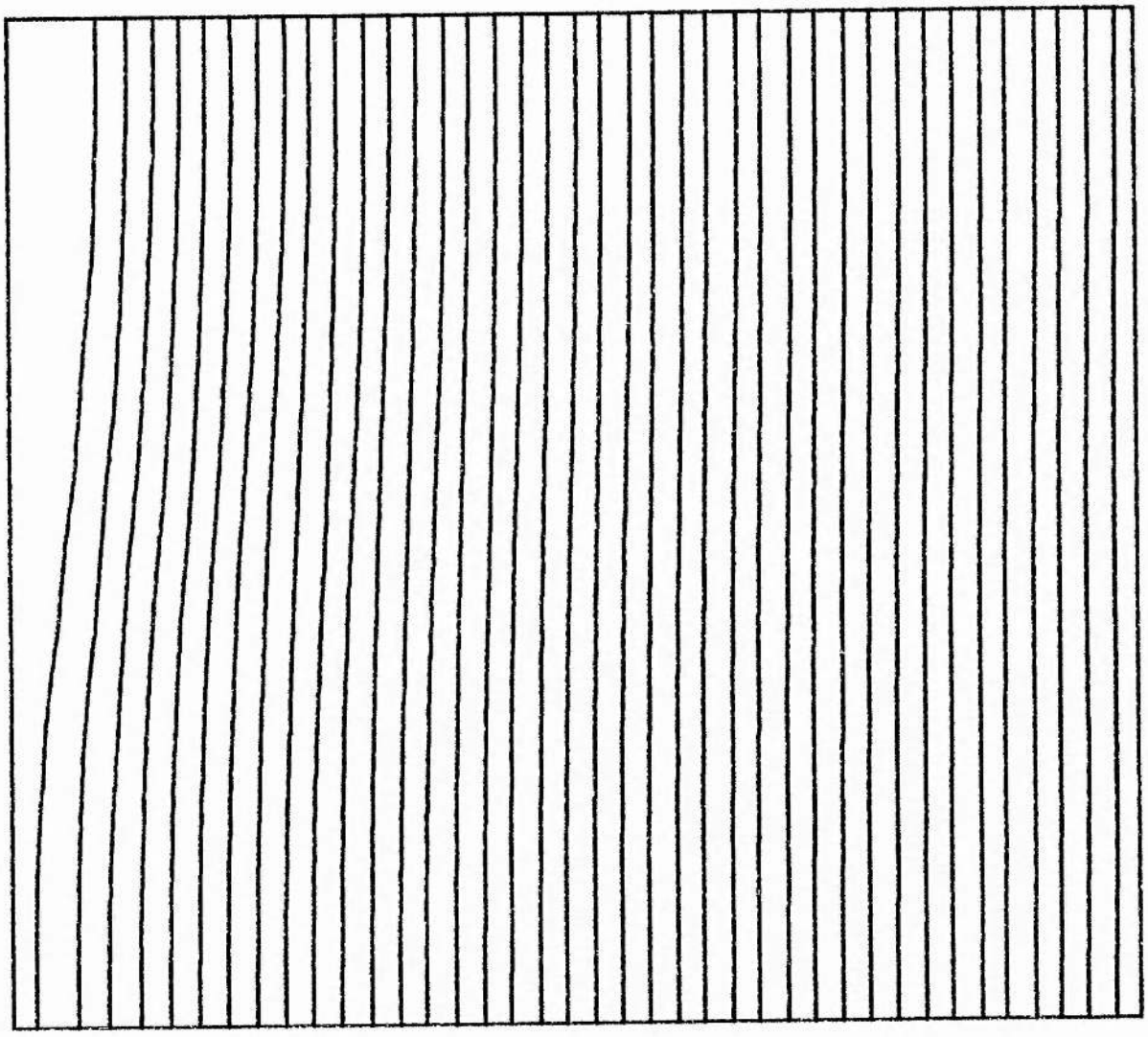


Figure 4.I0 Magnetic field lines

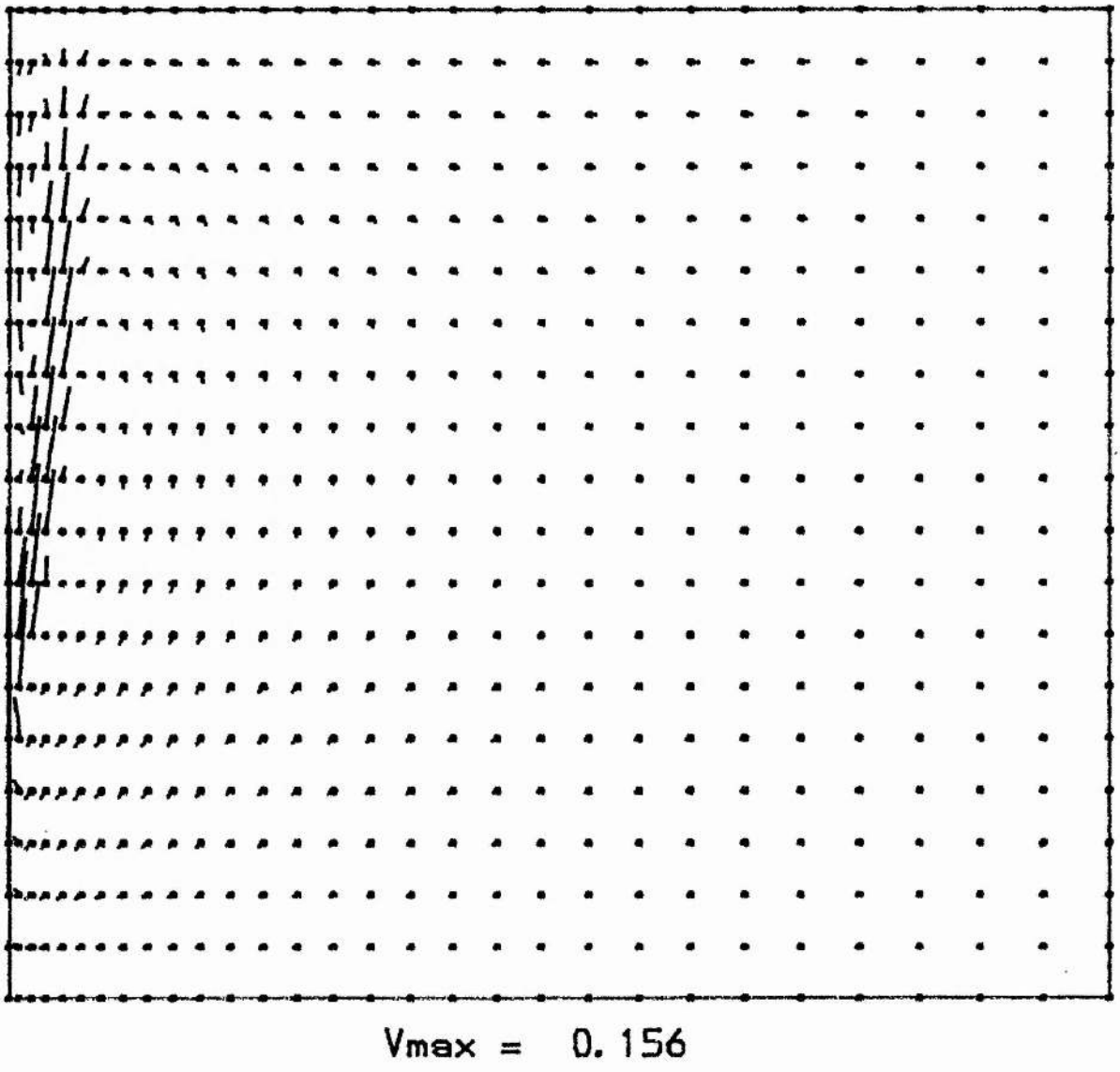


Figure 4.II Plasma flow vectors

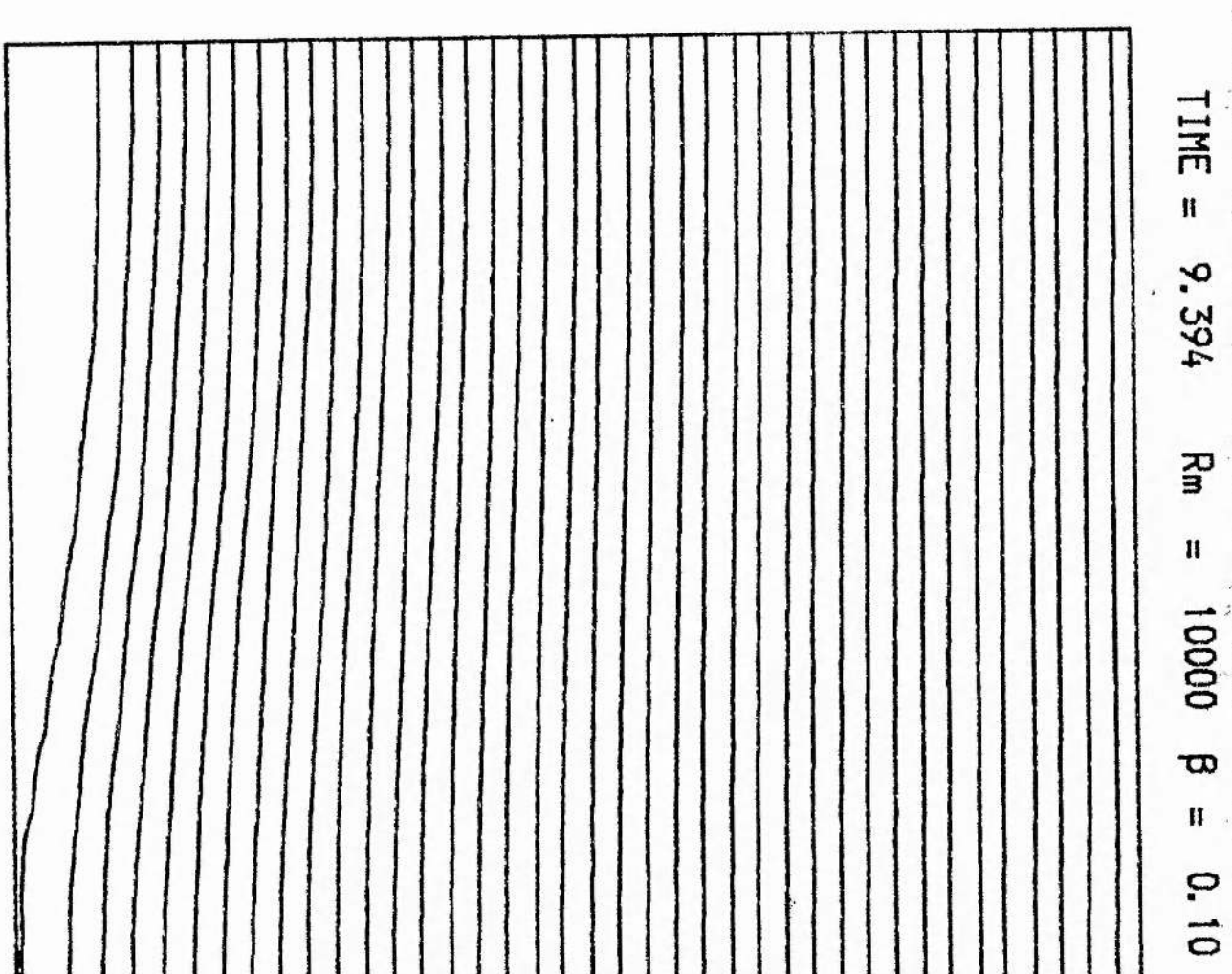


Figure 4.I2 Magnetic field lines

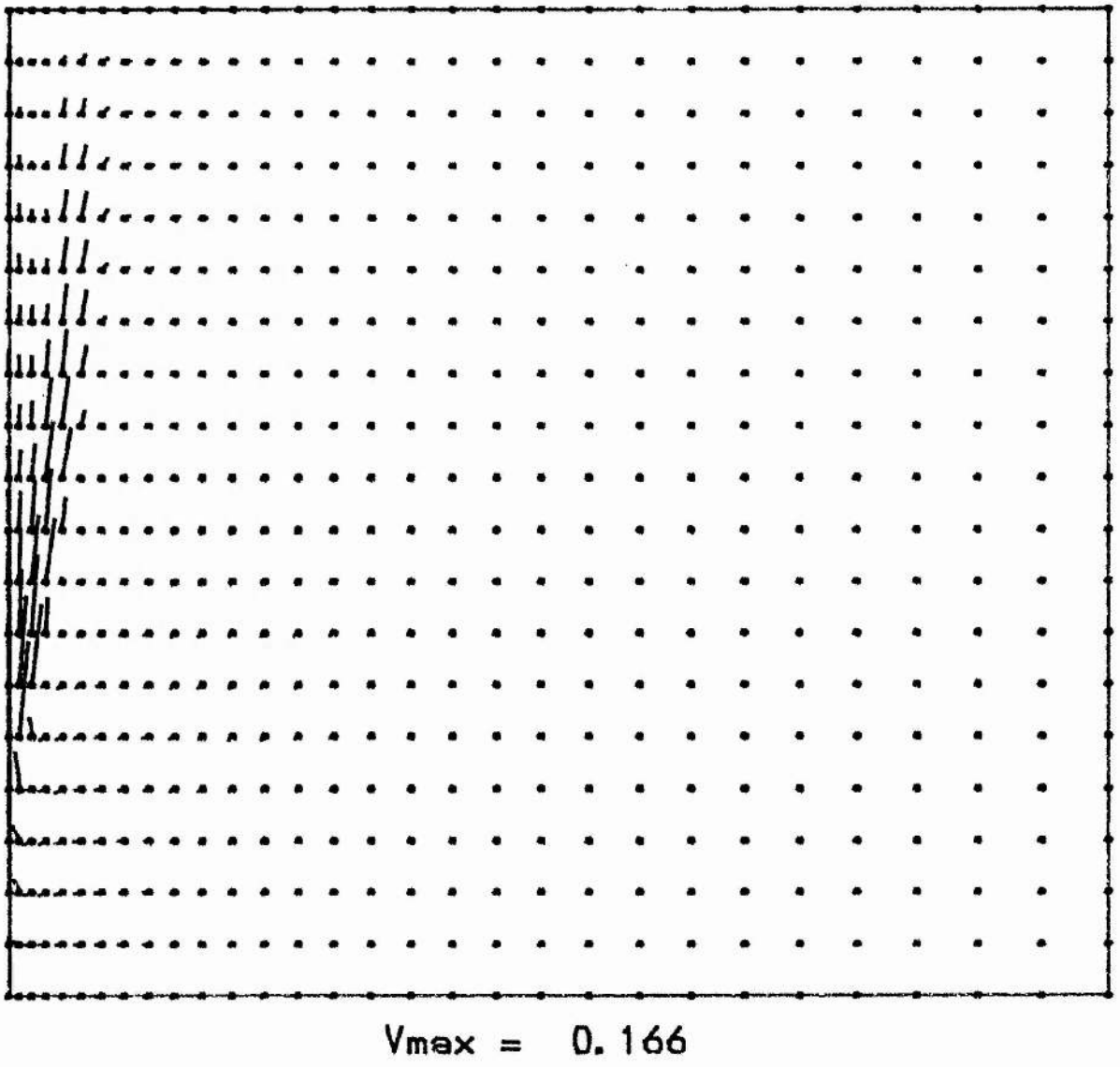


Figure 4.I3 Plasma flow vectors

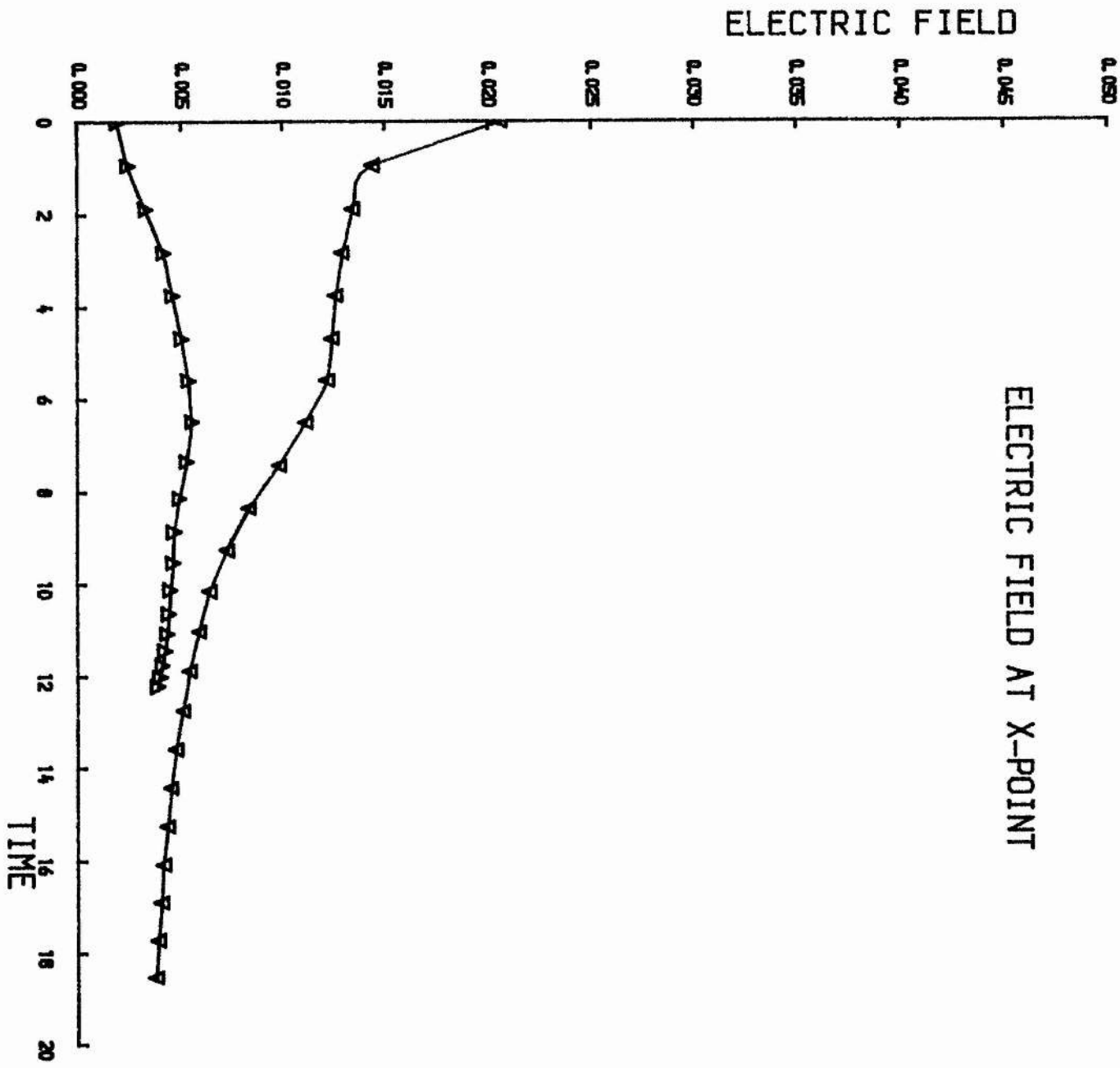


Figure 4.I4 Electric field at X-point for different R_m; open-ended reconnection

∇: R_m = I000

Δ: R_m = I0000

Initial perturbation level = 0.0I

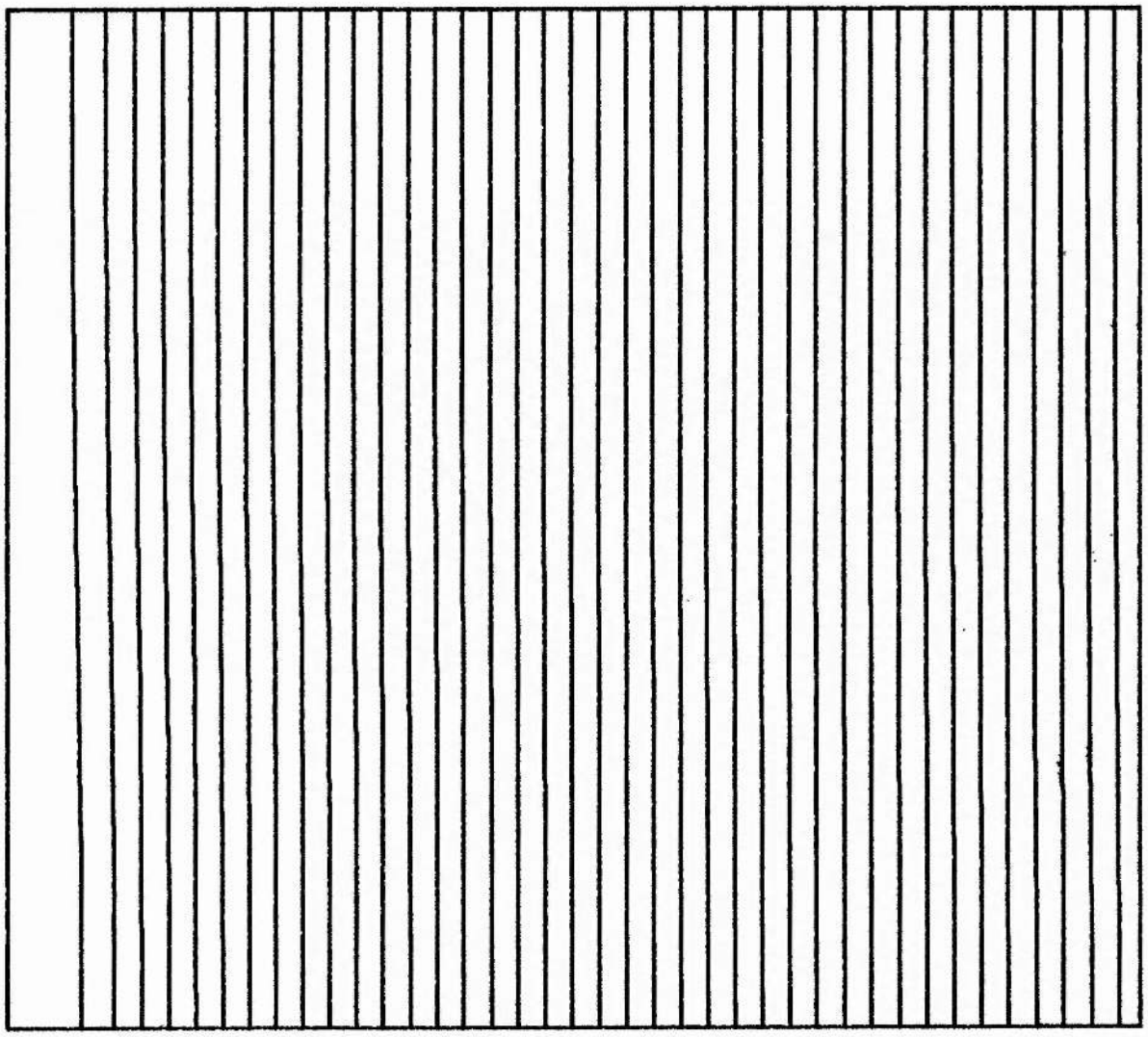


Figure 4.I5 Magnetic field lines

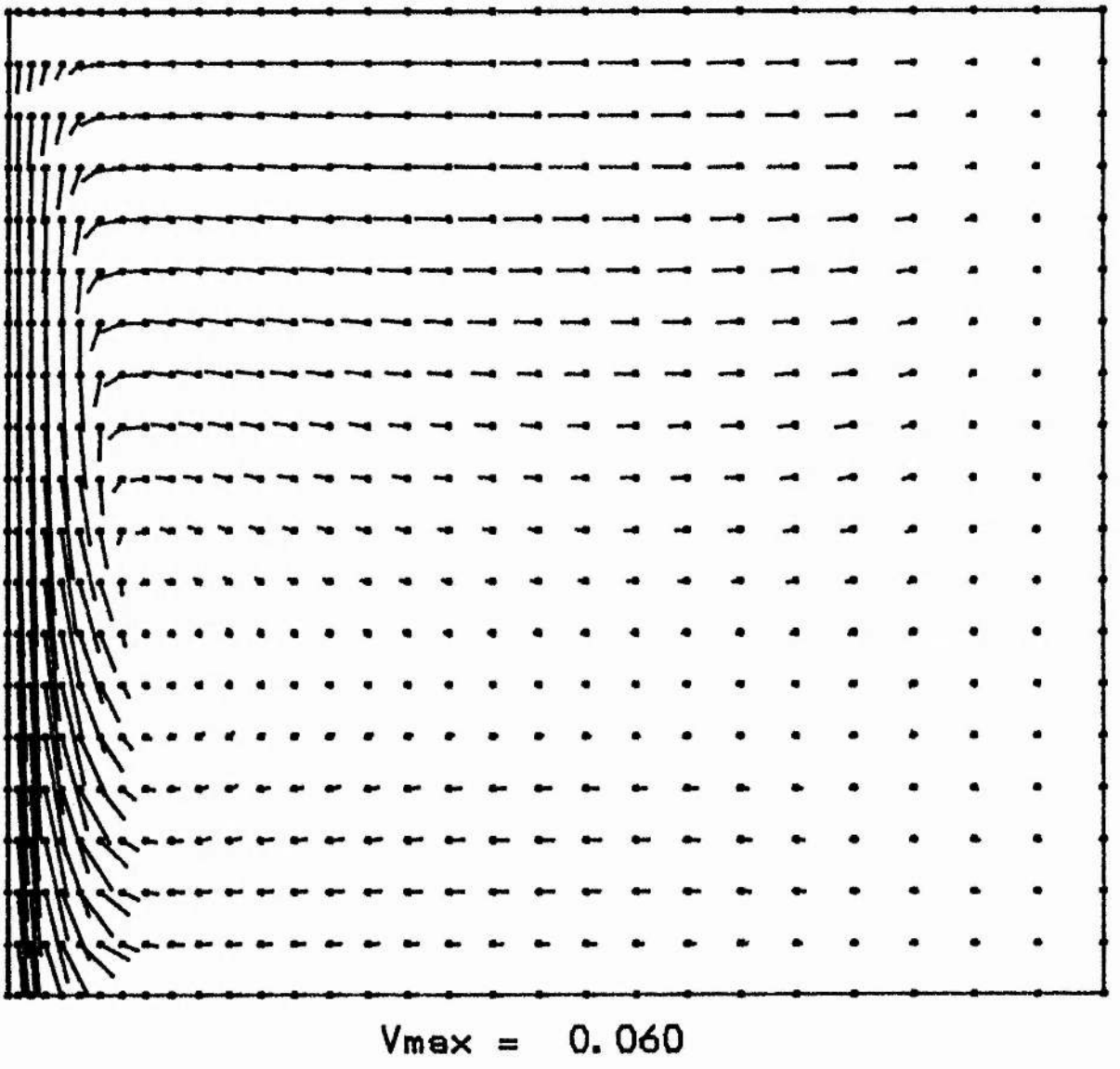


Figure 4.I6 Plasma flow vectors

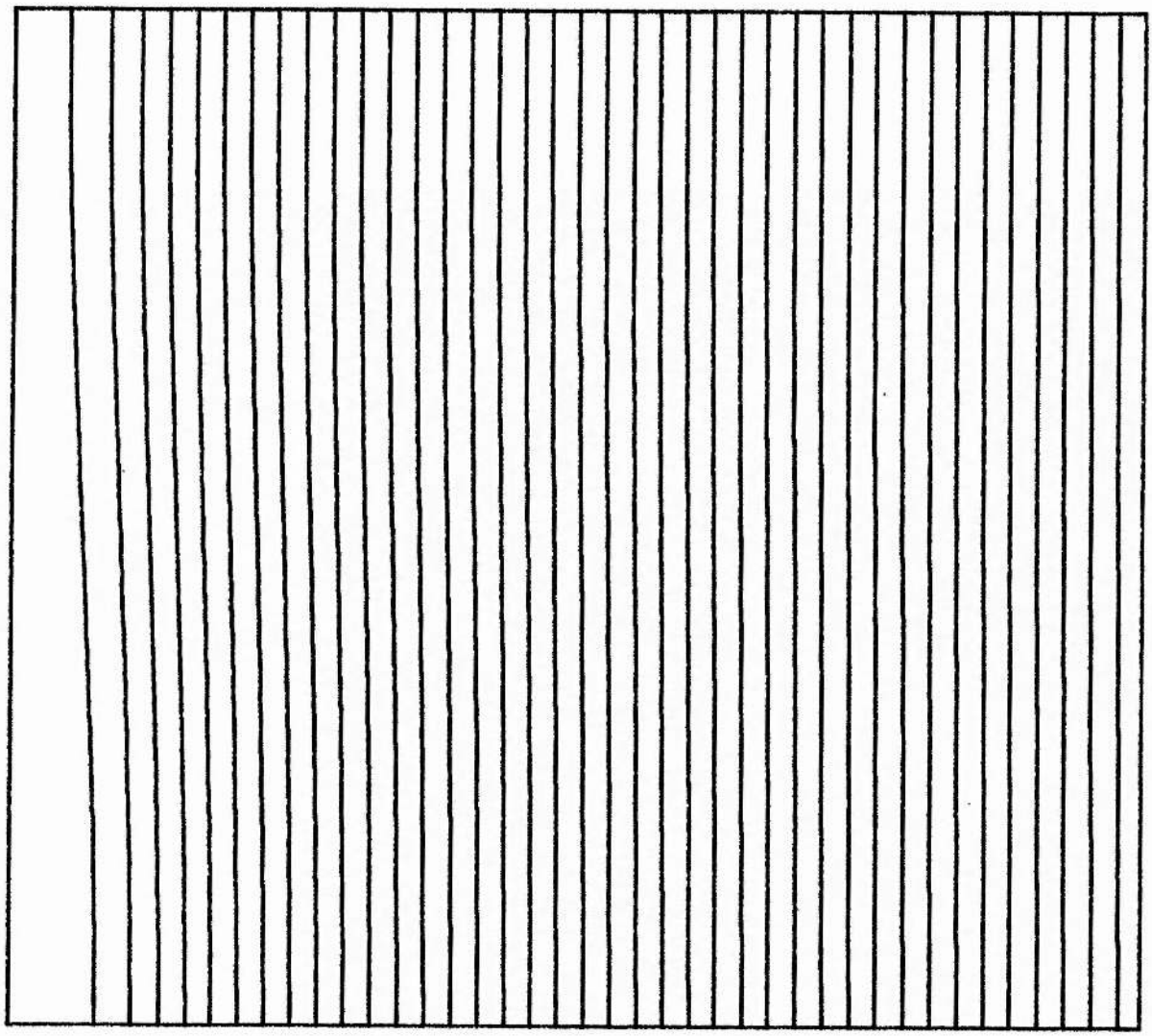


Figure 4.17 Magnetic field lines

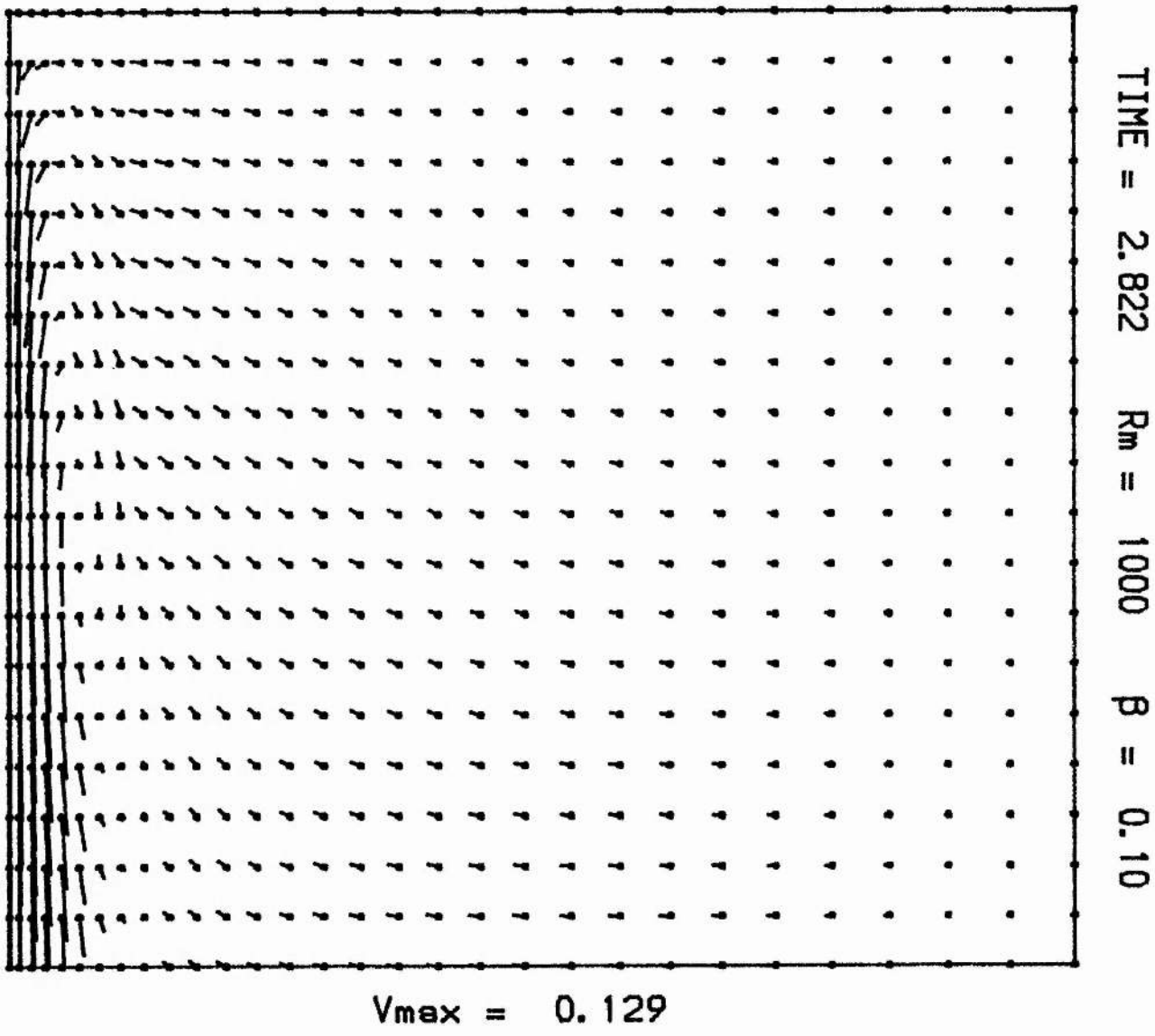


Figure 4.I8 Plasma flow vectors

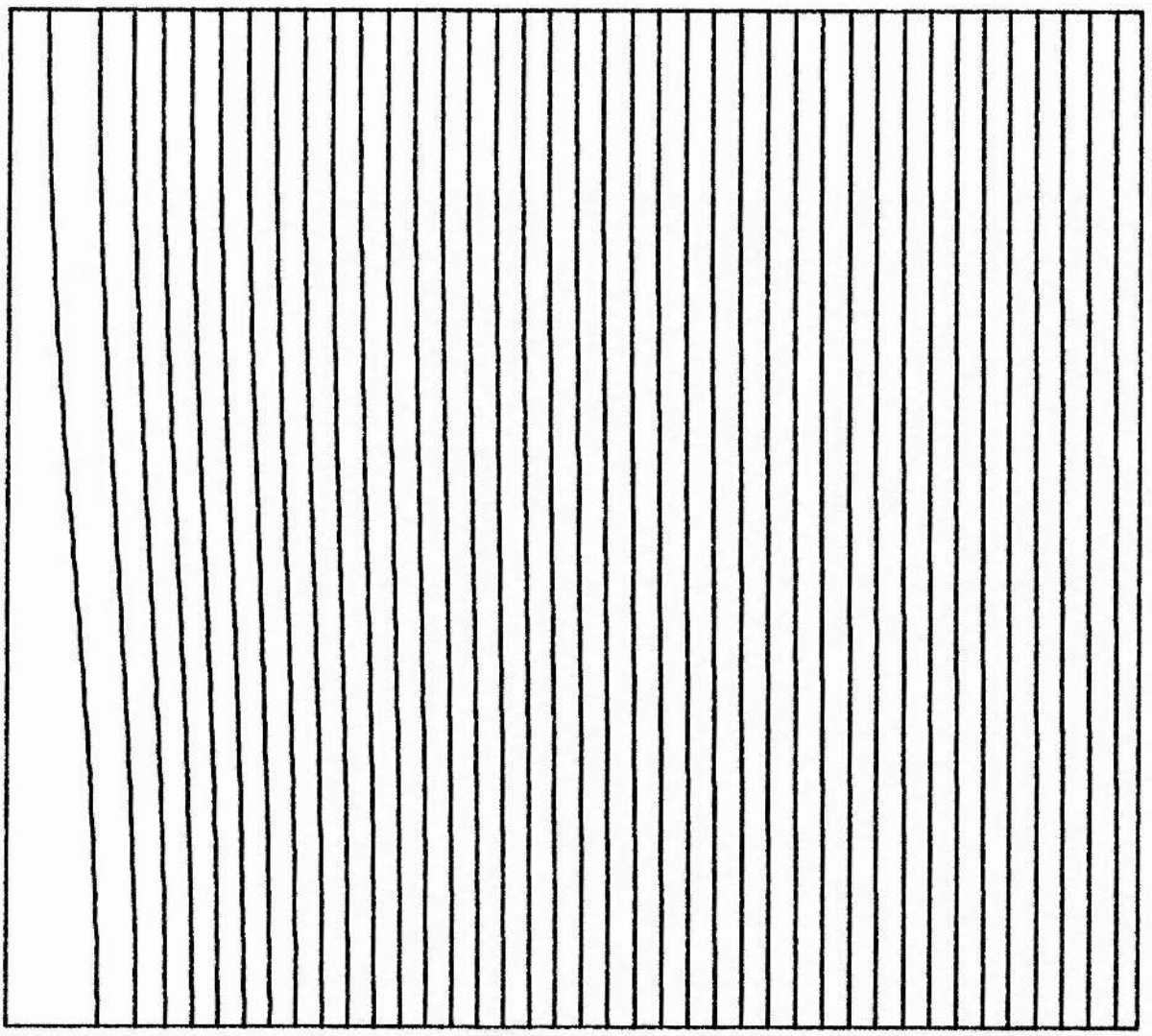


Figure 4.19 Magnetic field lines

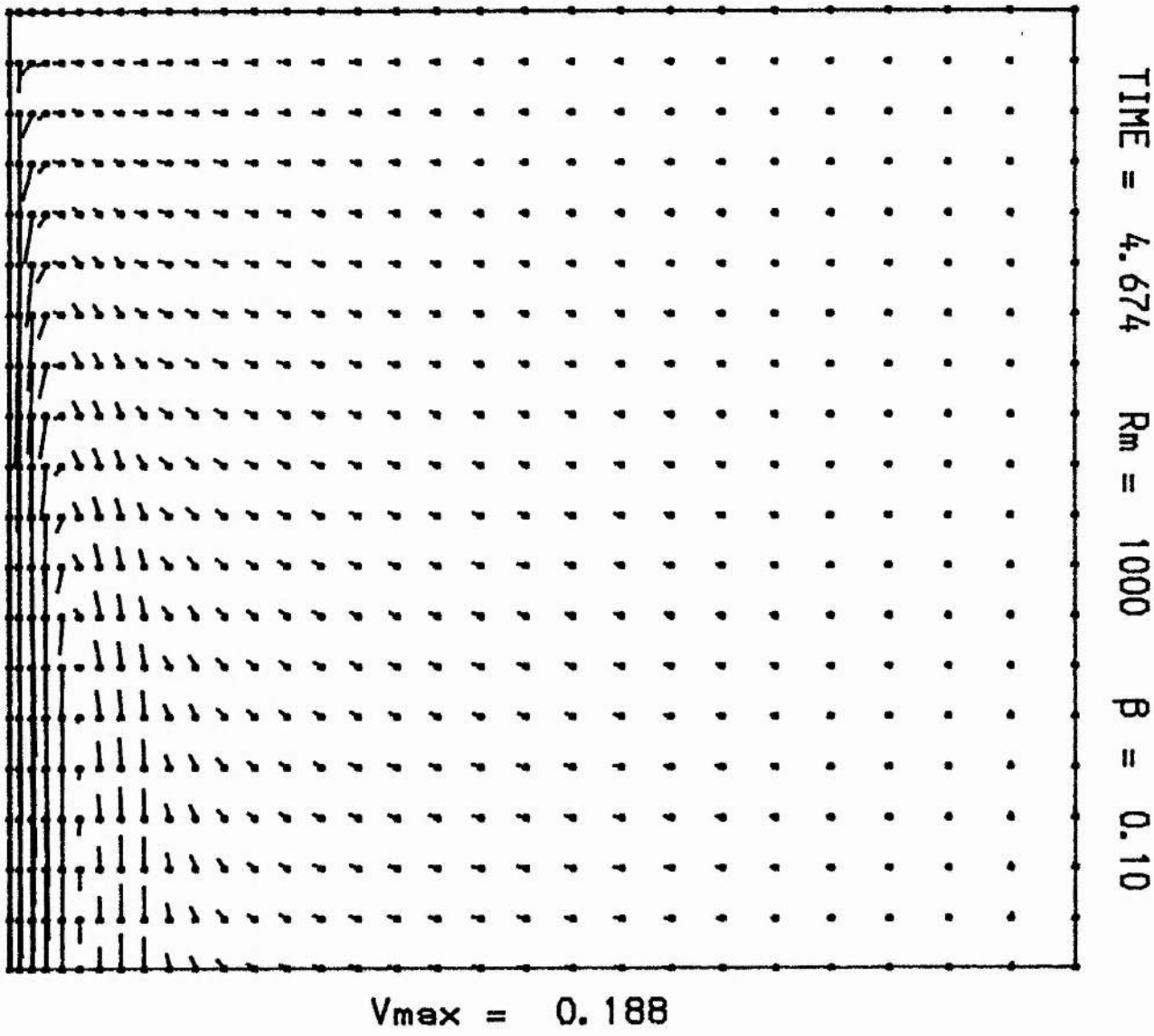


Figure 4.20 Plasma flow vectors

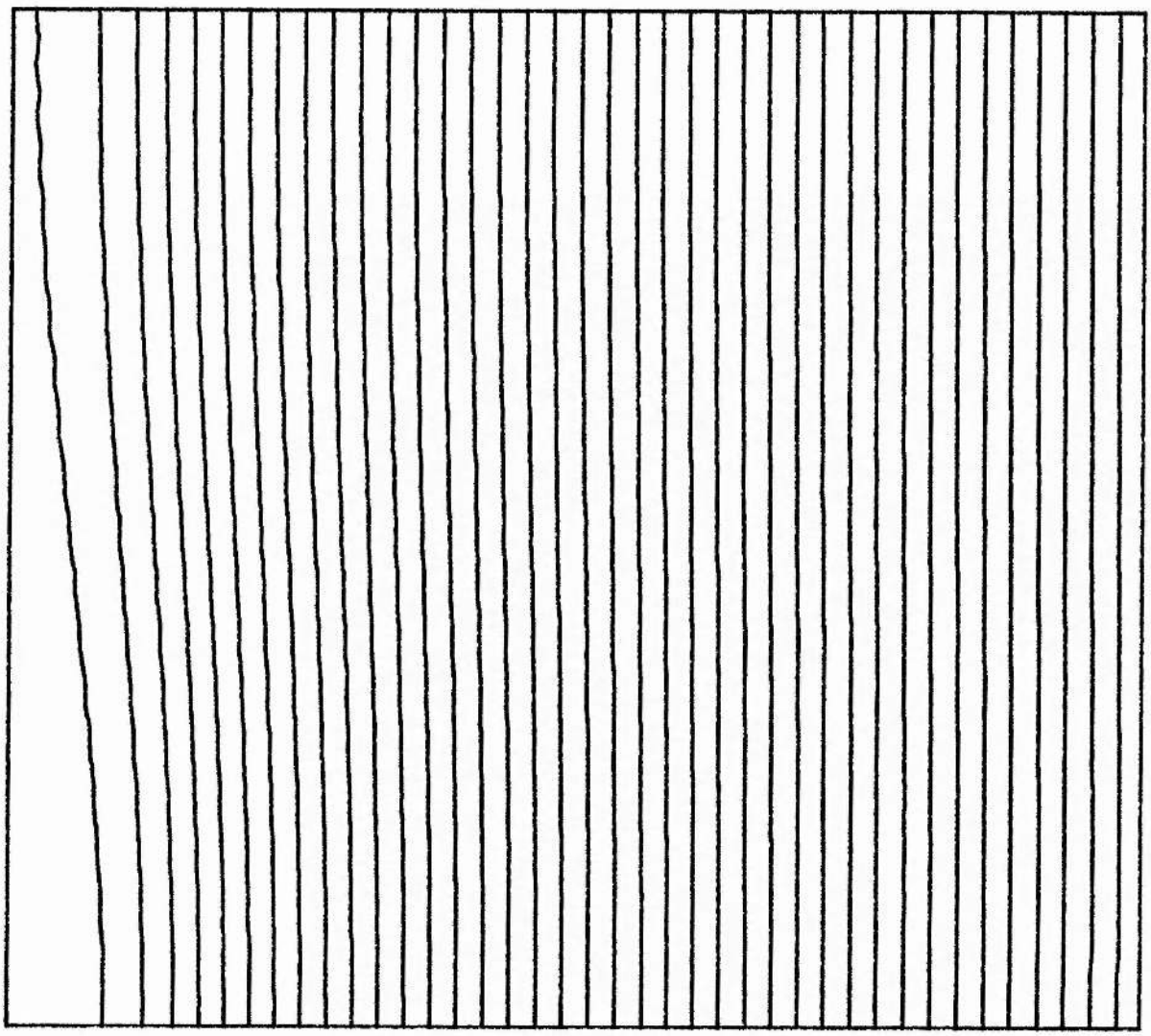


Figure 4.2I Magnetic field lines

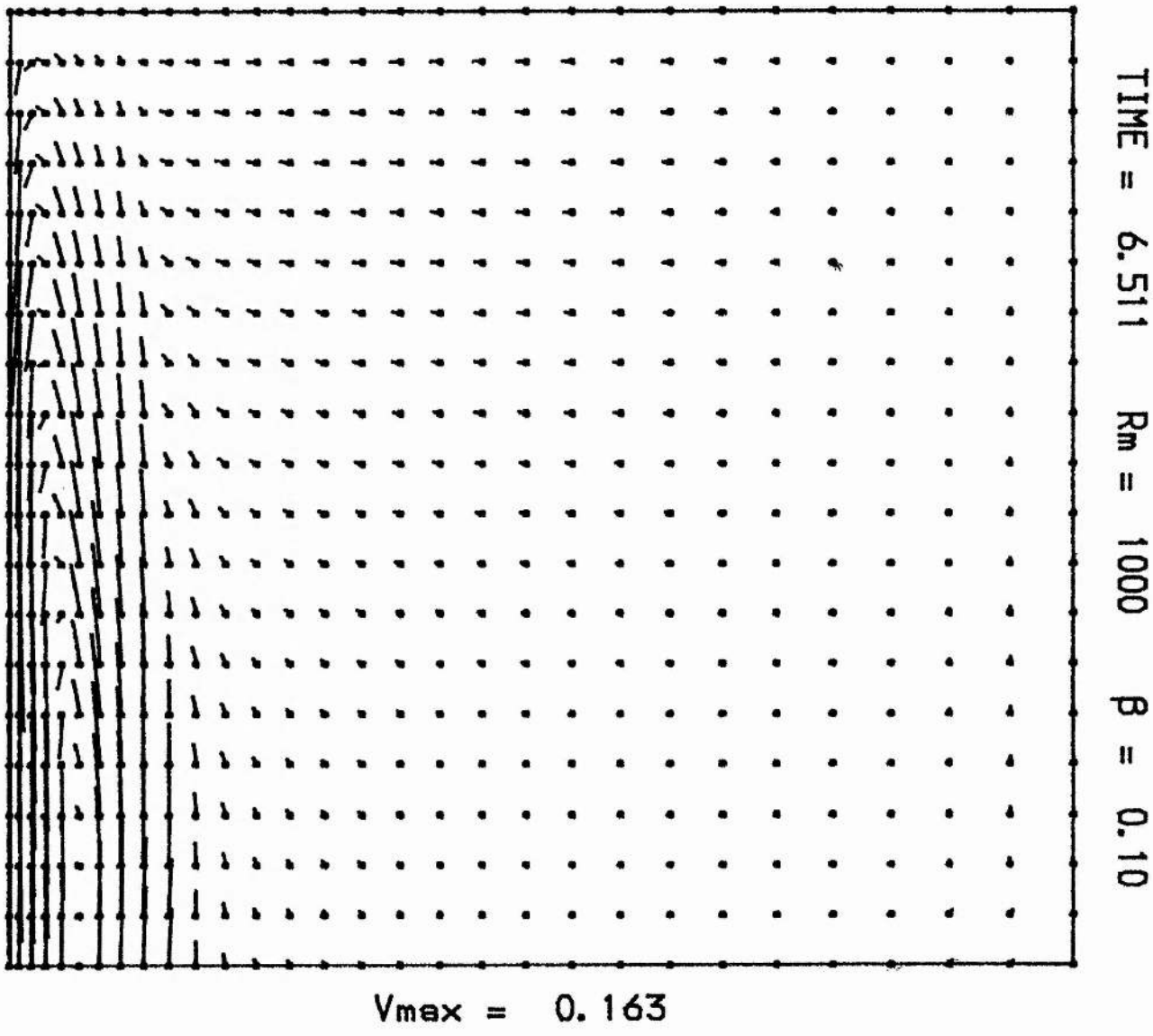


Figure 4.22 Plasma flow vectors

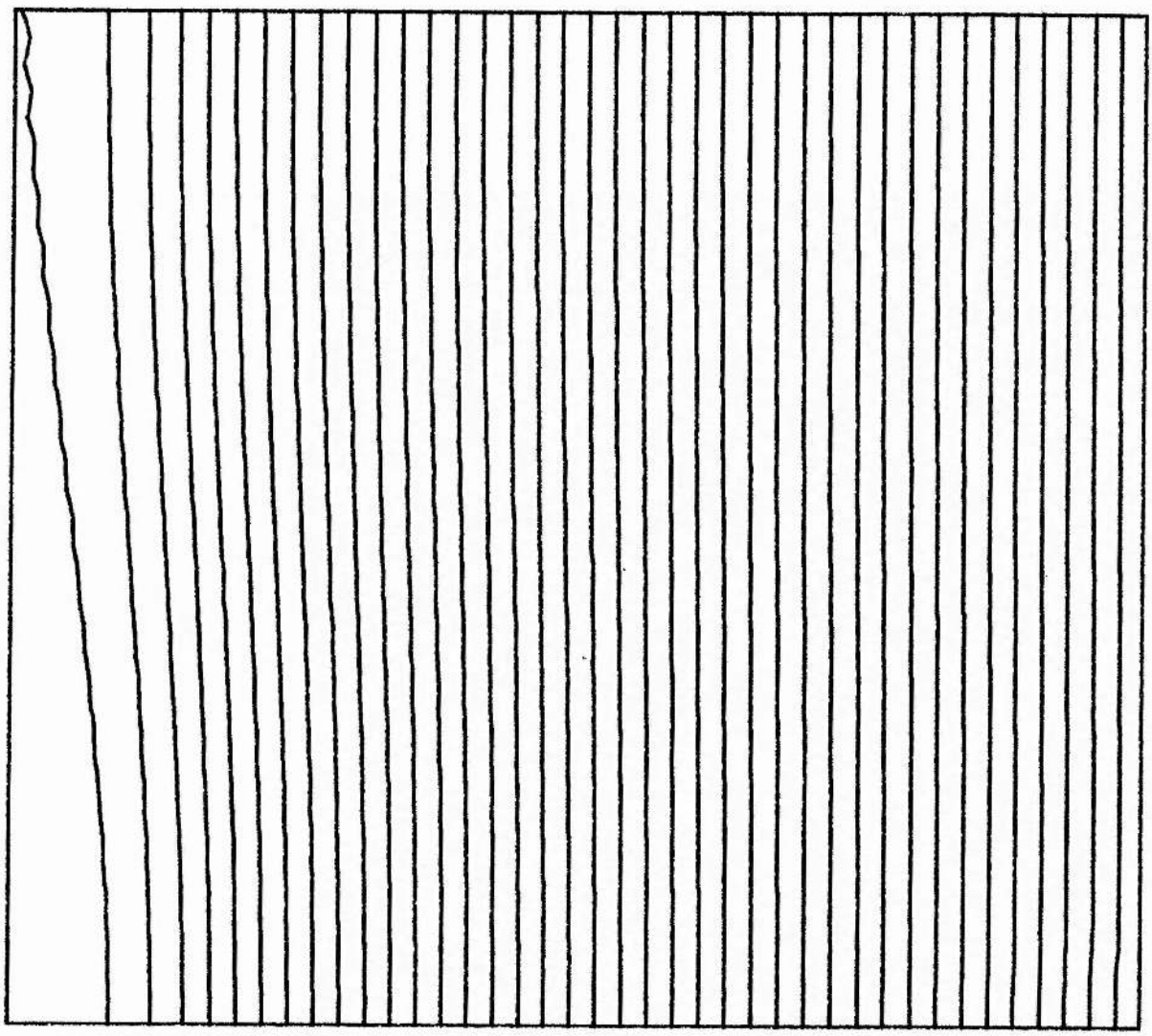


Figure 4.23 Magnetic field lines

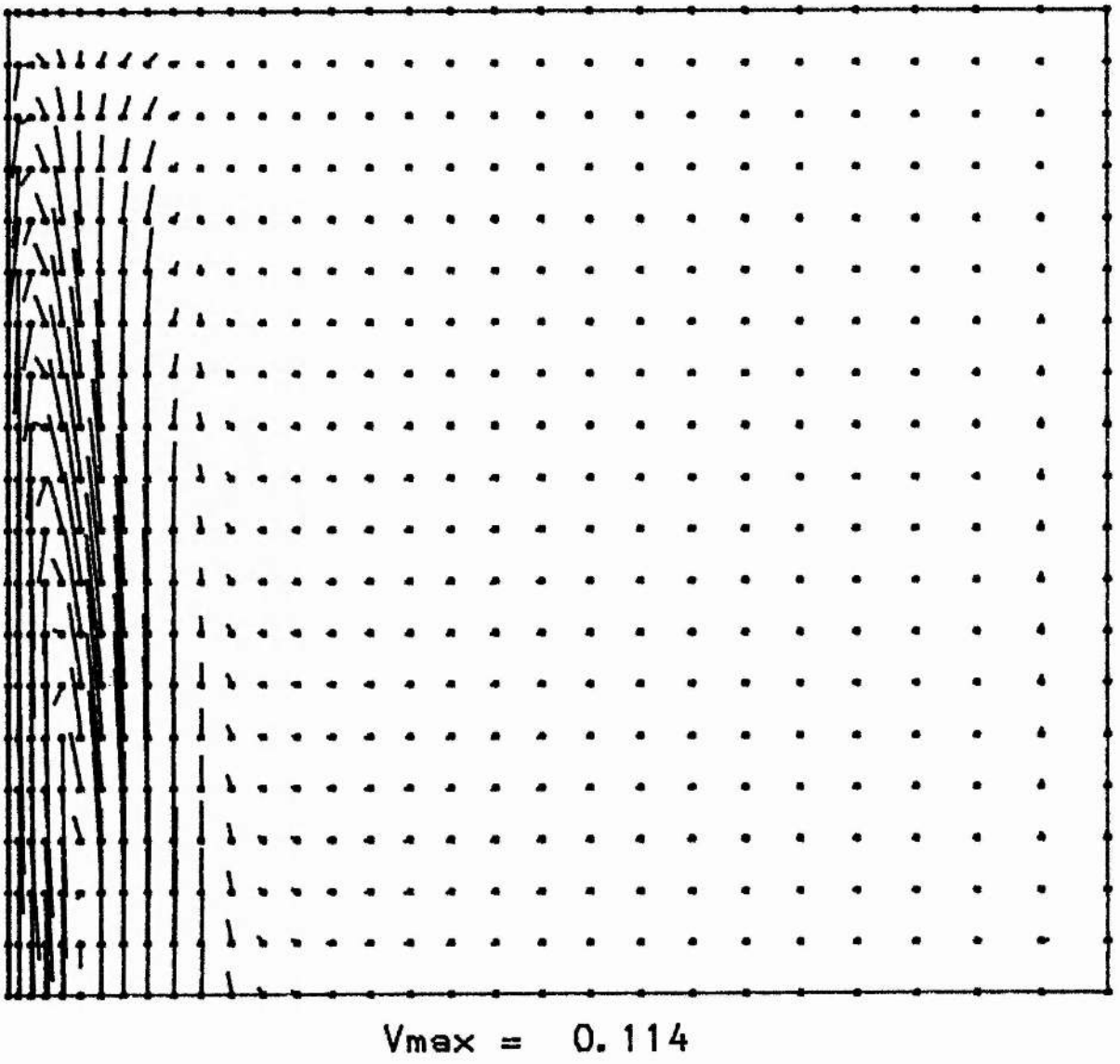


Figure 4.24 Plasma flow vectors

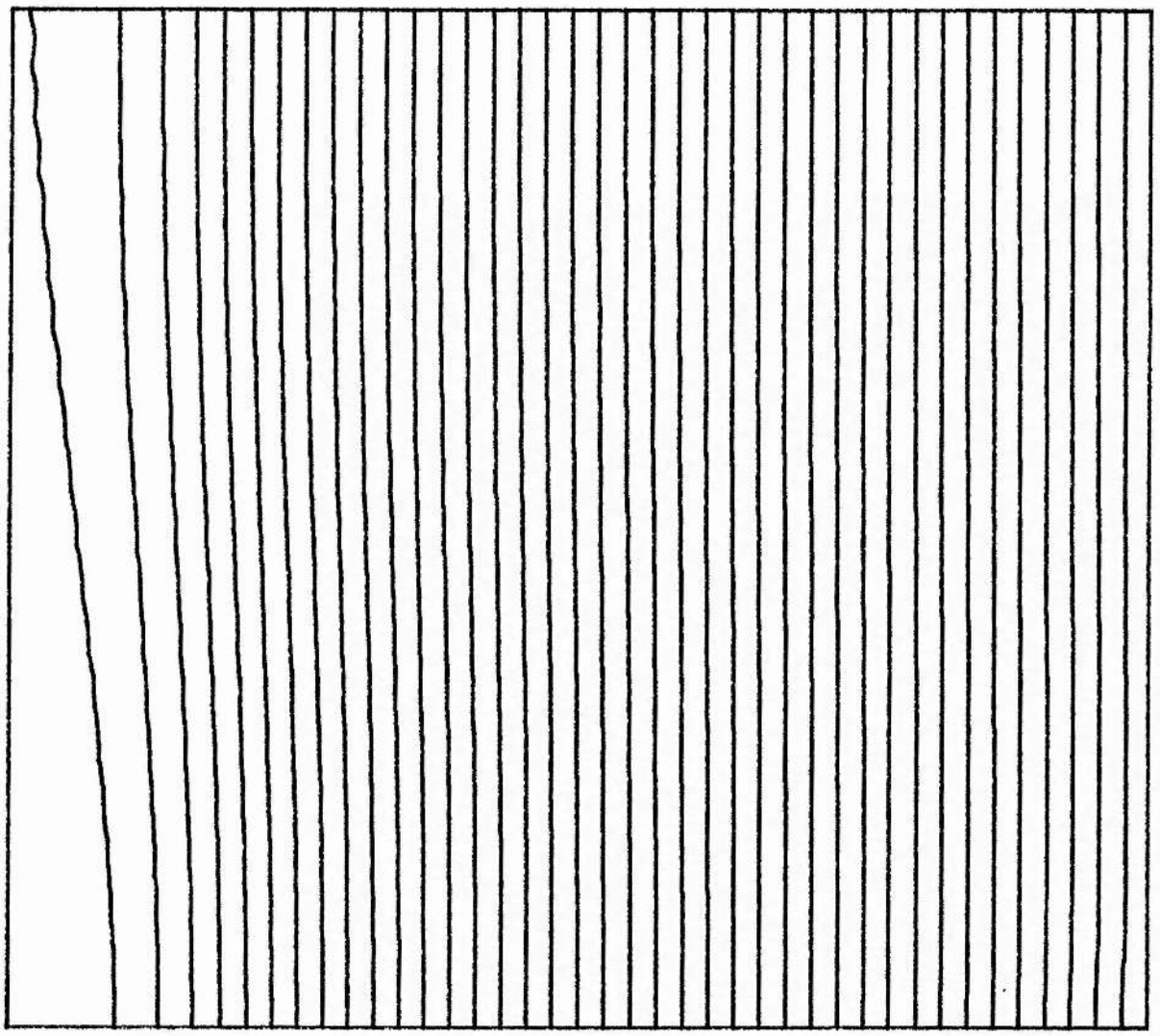


Figure 4.25 Magnetic field lines

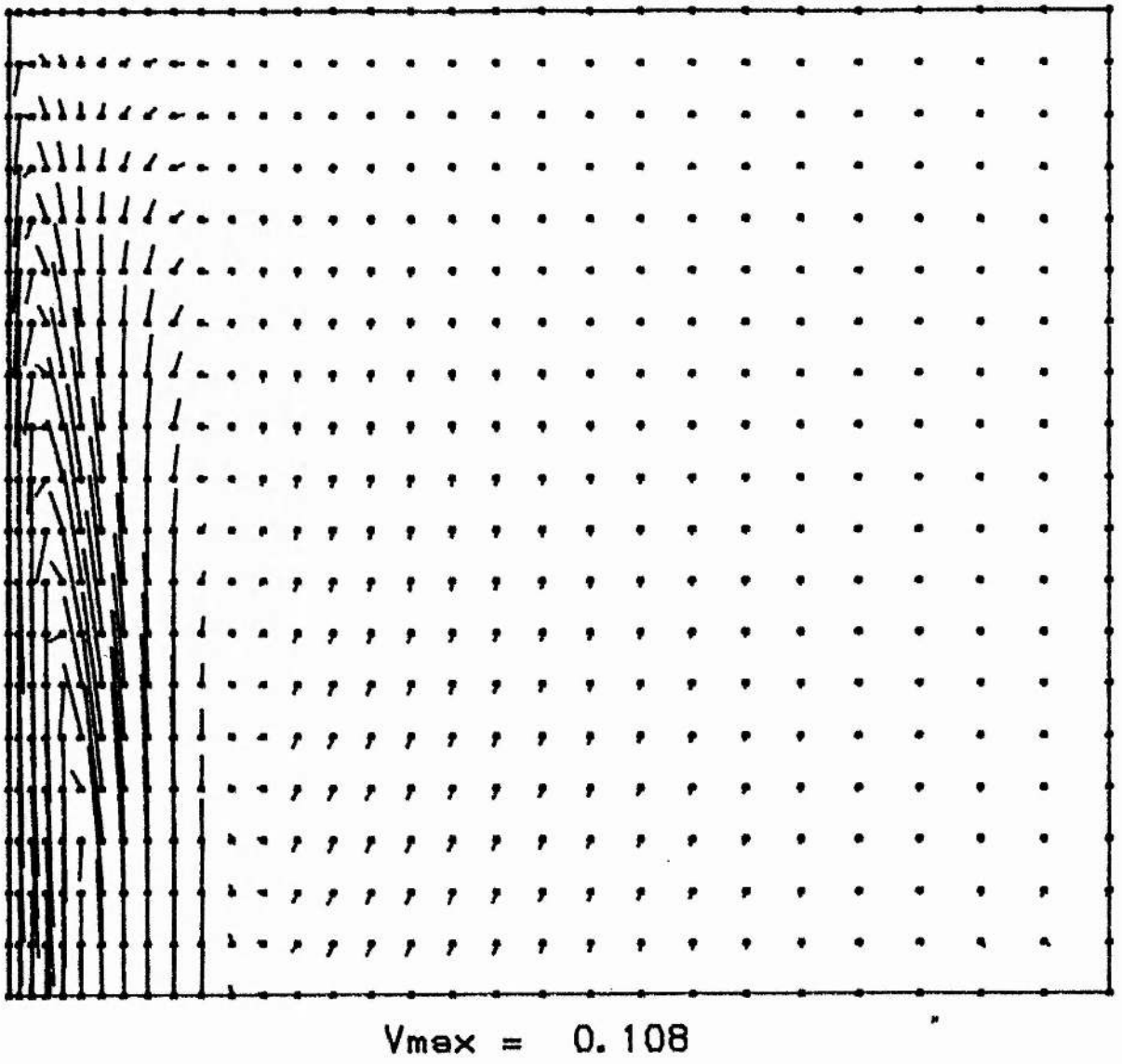


Figure 4.26 Plasma flow vectors

CHAPTER 5

ISLAND COALESCENCE

5.1 Introduction

The formation of magnetic islands is a consequence of non-linear saturation of the tearing mode instability. In the last decade much effort has been put into documenting their occurrence in laboratory plasmas, especially since magnetic tearing is conjectured as a major cause of disruptions in tokamaks and reverse field pinches. It is well known that a current layer separating two regions of plasma with oppositely oriented magnetic fields is unstable to a tearing mode, forming a periodic two-dimensonal magnetic configuration. Observations of tokamak discharges at later times show an absence of high m, n helical modes, and several workers have suggested that this is because the associated high wavenumber islands have coalesced together. This energetically favourable process would continue until the minimum m, n values are reached consistent with the helicity constraints of the particular rational surface about which tearing has occurred.

Finn and Kaw (1977) were the first to perform a mathematical analysis of the coalescence instability by studying the incompressible, one-fluid MHD equations in the x-y plane: in a tokamak, the z-coordinate follows the direction of helical symmetry. Incompressibility is justified in both the high and low β regimes; for $\beta \gg 1$ (as in some astrophysical plasmas) the sound wave timescale is much greater than all others so that density perturbations are negligible, while for $\beta \ll 1$ (as in a tokamak) the high toroidal magnetic field prevents lateral compression of the plasma.

Finn and Kaw start with the assumption of an ideal hydromagnetic equilibrium, so that the magnetic vector potential ψ_0 satisfies

$$\nabla^2 \psi_0 = f(\psi_0) \tag{5.1.1}$$

$$\underline{B}_0 = -B_p \hat{\underline{e}}_z \times \underline{\nabla} \psi_0 \tag{5.1.2}$$

for an arbitrary functional form f (Finn and Kaw, 1977). They generalise the 'tanh profile'

$$B_{0y} \sim \tanh x \tag{5.1.3}$$

to two dimensions, namely

$$\psi_0(x,y) = \log_e (\cosh kx + \varepsilon \cos ky) \tag{5.1.4}$$

The current distribution for an arbitrary flux function is given by

$$J(\psi) = -\nabla^2 \psi \tag{5.1.5}$$

implying in this case

$$J(\psi_0) = -(1-\varepsilon^2) k^2 \exp(-2\psi_0) \tag{5.1.6}$$

The incompressibility of the plasma allows a velocity potential $\phi(x,y)$ to be defined, where

$$\underline{v} = \hat{\underline{e}}_z \times \underline{\nabla} \phi \qquad (5.1.7)$$

Perturbations of interest force neighbouring islands together, so a velocity potential is chosen that makes the y-component of velocity zero at the x points:

$$\phi(x,y) = \sum_{i=0}^{\infty} \phi_i(x) \sin \frac{(2i-1)ky}{2} \qquad (5.1.8)$$

The functions ϕ_i are odd in x and are localised near the island (near $|kx| \leq \varepsilon^{\frac{1}{2}}$). An energy principle analysis is performed with three harmonics, in which (5.1.8) is approximated by

$$\phi(x,y) = \phi_0(x)\left[\sin\frac{y}{2} + \alpha \sin\frac{3y}{2} + \beta \sin\frac{5y}{2}\right] \qquad (5.1.9)$$

$$\phi_0(x) = \tanh^2(\lambda \varepsilon^{-\frac{1}{2}} x) e^{-\frac{1}{2}x}(2 + \coth x) \qquad (5.1.10)$$

and the energy integral is evaluated while ε is held constant and α, β, γ varied. With α, β nonzero δW is minimised at $\lambda \approx 1$, $\alpha = -0.32$, $\beta = 0.1$, $\epsilon > 0.06$. This leads the authors to suggest an 'ϵ-threshold' for coalescence; the instability is only seen when the islands grow to a certain width.

The same problem is treated purely numerically by Pritchett and Wu (1979). Starting with a chain of two islands, they advance the 2-D incompressible MHD equations forward in time with a variety of different initial conditions. The coalescence instability is always seen but no threshold is observed. With the inclusion of resistivity terms in the

equations the two islands merge into one. The same result has been confirmed independently by Biskamp and Welter (1980).

This work was extended by Pellat (1983). He takes an energy integral of the incompressible MHD equations and uses the trial function

$$\psi(x,y) = \alpha\left(f_{\frac{1}{2}}(x)\cos\frac{ky}{2} + f_{\frac{3}{2}}(x)\cos\frac{3ky}{2}\right) \quad (5.1.9)$$

to show that

$$\delta W = -\frac{\pi\alpha^3}{3} - \frac{8\alpha}{3} + O(\epsilon) \quad (5.1.10)$$

which is always negative, so there is no threshold to a coalescence instability.

Hayashi (1981) performs a similar study to Pritchett and Wu but takes different initial conditions. Instead of assuming an *a priori* island structure, he perturbs a current sheet so that it tears, and tests the stability of the resulting islands which have a wavenumber k = 0.5 in his non-dimensionalised units. In contrast to the other results he finds that the islands are strongly stable, even when fairly large perturbations are applied (e.g. an imposed flow of up to 20% of the external Alfven speed).

In this chapter we present some results that reconcile these seemingly contradictory findings. Firstly, we examine the magnetic field profile generated by a tearing mode when in the linear, exponentially growing phase. Such a configuration arises when a component of magnetic field changes sign over some surface in a plasma; then with even a very slightly resistive medium the field lines 'tear' and through

reconnection form a chain of magnetic islands. Further details are given in the classic paper by Furth, Killeen and Rosenbluth (1963). We then show that Hayashi's observation may be justified mathematically because of a fundamental difference in the nature of the tearing mode and the Finn and Kaw flux functions. The previous work is extended to a more realistic magnetic field profile for tearing modes. We demonstrate that under certain conditions there does after all exist a threshold for island coalescence, but that it is a threshold in wavenumber rather than island width.

It appears to be a common misconception in the literature that the Finn and Kaw profile represents the saturated state of a tearing mode, and that an investigation of the ideal stability of this profile is applicable to tearing modes. Hayashi observes that this is not necessarily so. In the report on his numerical simulation, he shows that while j_z produced by tearing has a maximum at the X-point and a minimum at the O-point, the *opposite* occurs in Finn and Kaw's initial state. We are led to ask: what are the stabilty criteria for a chain of islands produced by a tearing mode?

This is a rather more complex problem than the previous one, for we have no general ψ_0 to represent a saturated tearing mode. However, rather than considering a mode that has entered the Rutherford regime or even saturated, it is probably more realistic to restrict any stability analysis to the linear phase. Tearing develops on a timescale $(\tau_A \tau_R)^{\frac{1}{2}}$ intermediate between the resistive (τ_R) and ideal (τ_A) timescales, while coalescence is an ideal instability. Therefore coalescence will occur very soon after island formation for S-values representative of conditions inside a fusion reactor. This will preclude further tearing.

The ideal stability analysis below uses a 'snapshot' of the slowly growing tearing mode in the exponentially growing phase as a starting point.

5.2 Derivation of the energy integral

To derive an energy integral in a suitable form for further analysis we use a form of Bernstein's (1957) original integral, first presented by Newcomb (1960):

$$\delta W = \frac{1}{2}\int [\underline{\nabla}\times(\underline{\xi}\times\underline{B}) - \underline{\nabla}\times(\underline{\xi}\times\underline{B})\cdot(\underline{\xi}\times\underline{j})]\, d^3r \qquad (5.2.1)$$

We omit the pressure terms which occur in the original expression because they involve a divergence of the perturbation $\underline{\xi}$, which is assumed to be zero from the incompressibility assumption ($\underline{\nabla}\cdot\underline{\xi} = 0$). Here

$$\underline{B} = \underline{\nabla}\psi \times \hat{\underline{e}}_z \quad \underline{j} = -\nabla^2\psi\,\hat{\underline{e}}_z \quad p = p(\psi) \quad -\nabla^2\psi = \frac{\partial p}{\partial \psi} \qquad (5.2.2)$$

Consider the two terms in the integral separately. For the first,

$$\underline{\nabla}\times(\underline{\xi}\times\underline{B}) = \underline{\nabla}\times(\xi_z\underline{\nabla}\psi - \underline{\xi}\cdot\underline{\nabla}\psi\,\hat{\underline{e}}_z) \qquad (5.2.3)$$

$$= \underline{\nabla}\xi_z\times\underline{\nabla}\psi - \underline{\nabla}\xi\times\hat{\underline{e}}_z \qquad (\xi = \underline{\xi}\cdot\underline{\nabla}\psi) \qquad (5.2.4)$$

and so

$$|\underline{\nabla}\times(\underline{\xi}\times\underline{B})|^2 = |\underline{\nabla}\xi_z\times\underline{\nabla}\psi|^2 + |\underline{\nabla}\xi|^2$$

$$= |\nabla \xi|^2 \qquad (5.2.5)$$

since $\xi_z = 0$, as the perturbation is assumed to be two dimensional.

To evaluate the second term we use the vector identity for the curl of a vector product, and note that

$$\nabla \times (\underline{\xi} \times \underline{B}) \cdot (\underline{\xi} \times \underline{j}) = \begin{matrix} \nabla \cdot [(\underline{\xi} \times \underline{B}) \times (\underline{\xi} \times \underline{j})] + \\ \nabla \times (\underline{\xi} \times \underline{j}) \cdot (\underline{\xi} \times \underline{B}) \end{matrix} \qquad (5.2.6)$$

The first term on the right is zero due to the simultaneous appearance of the two $\underline{\xi}$ terms. For the second, we examine

$$\nabla \times (\underline{\xi} \times \underline{j}) = \nabla \times (\underline{\xi} \times \frac{\partial p}{\partial \psi} \hat{\underline{e}}_z) \qquad (5.2.7)$$

$$= (\underline{\xi} \cdot \nabla) \frac{\partial p}{\partial \psi} \hat{\underline{e}}_z \qquad (5.2.8)$$

$$= \underline{\xi} \cdot \nabla \psi \frac{\partial^2 p}{\partial \psi^2} \hat{\underline{e}}_z \qquad (5.2.9)$$

$$= \xi \frac{\partial^2 p}{\partial \psi^2} \hat{\underline{e}}_z \qquad (5.2.10)$$

where $\xi = \underline{\xi} \cdot \nabla \psi \qquad (5.2.11)$

Then $\nabla \times (\underline{\xi} \times \underline{j}) \cdot (\underline{\xi} \times \underline{B}) = -\xi^2 \frac{\partial^2 p}{\partial \psi^2} \qquad (5.2.12)$

and so

$$\delta W = \frac{1}{2} \int \left[|\nabla \xi|^2 - \frac{\partial^2 p}{\partial \psi^2} \xi^2 \right] d^2 r$$

$$= \frac{1}{2} \int \left[|\nabla \xi|^2 - \frac{\partial J}{\partial \psi} \xi^2 \right] d^2 r$$

5.3 Minimisation of the energy integral

We require a perturbation ξ that minimises δW subject to the normalisation constraint

$$\int_\Pi \xi^2 \, dS = 1 \qquad (5.3.1)$$

Since δW is of the form

$$\int_\Pi f(x, y, \xi, \frac{\partial \xi}{\partial x}, \frac{\partial \xi}{\partial y}) \, dx \, dy \qquad (5.3.2)$$

the Euler-Lagrange equation is

$$\frac{\partial f}{\partial \xi} - \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial \xi_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial \xi_y}\right) = 2\mu\xi \qquad (5.3.3)$$

where $\xi_x = \frac{\partial \xi}{\partial x}$ $\qquad$ $\xi_y = \frac{\partial \xi}{\partial y}$

which gives

$$-\xi \frac{\partial J}{\partial \psi} - \frac{\partial^2 \xi}{\partial x^2} - \frac{\partial^2 \xi}{\partial y^2} = \mu\xi \qquad (5.3.4)$$

yielding

$$-\nabla^2 \xi - \frac{\partial J}{\partial \psi}\xi = \mu\xi \qquad (5.3.5)$$

Because we have minimised subject to the constraint (5.3.1),

$$\delta W = \mu \tag{5.3.6}$$

5.4 Perturbation expansion of the Euler-Lagrange equation

We expand the quantities ξ, μ, $\frac{\partial J}{\partial \psi}$ and the initial profile ψ that occur in the Euler-Lagrange equations in powers of ε, where ε is a small parameter in the expression for J describing the departure from a one dimensional current sheet similar to that in (5.1.4). Thus

$$\xi = \xi_0 + \varepsilon \xi_1 + \varepsilon^2 \xi_2 + \varepsilon^3 \xi_3 + O(\varepsilon^4) \tag{5.4.1}$$

$$\mu = \mu_0 + \varepsilon \mu_1 + \varepsilon^2 \mu_2 + \varepsilon^3 \mu_3 + O(\varepsilon^4) \tag{5.4.2}$$

$$\frac{\partial J}{\partial \psi} = \frac{\partial J}{\partial \psi}\Big|_0 + \varepsilon \frac{\partial J}{\partial \psi}\Big|_1 + \varepsilon^2 \frac{\partial J}{\partial \psi}\Big|_2 + \varepsilon^3 \frac{\partial J}{\partial \psi}\Big|_3 + O(\varepsilon^4) \tag{5.4.3}$$

$$\psi = \psi_0 + \varepsilon \psi_1 + \varepsilon^2 \psi_2 + O(\varepsilon^3) \tag{5.4.4}$$

We define the operator

$$L(\chi) = -\nabla^2 \chi - \frac{\partial J}{\partial \psi}\Big|_0 \chi \tag{5.4.5}$$

To sucessive orders the Euler-Lagrange equation is then

$$O(\varepsilon^0): \quad L(\xi_0) = \mu_0 \xi_0 \tag{5.4.6}$$

$$O(\varepsilon^1): L(\xi_1) = \frac{\partial J}{\partial \psi}\Big|_1 \xi_0 + \mu_0 \xi_1 + \mu_1 \xi_0 \tag{5.4.7}$$

$$O(\varepsilon^2): L(\xi_2) = \frac{\partial J}{\partial \psi}\Big|_1 \xi_1 + \frac{\partial J}{\partial \psi}\Big|_2 \xi_0 + \mu_0 \xi_2 + \mu_1 \xi_1 + \mu_2 \xi_0 \tag{5.4.8}$$

$$O(\varepsilon^3): L(\xi_3) = \frac{\partial J}{\partial \psi}\Big|_1 \xi_2 + \frac{\partial J}{\partial \psi}\Big|_2 \xi_1 + \frac{\partial J}{\partial \psi}\Big|_3 \xi_0 + \mu_0 \xi_3 + \mu_1 \xi_2 + \mu_2 \xi_1 + \mu_3 \xi_0 \tag{5.4.9}$$

This system of equations is solved subject to a set of boundary conditions which are derived below.

The reader should note that two different orderings have been tacitly assumed in the above expansions. Denoting the total, perturbed flux function by A, we have

$$A = \psi + \delta\xi + O(\delta^2) \tag{5.4.10}$$

$$= \psi + \delta(\xi_0 + \varepsilon\xi_1 + O(\varepsilon^2)) + O(\delta^2) \tag{5.4.11}$$

implying

$$\delta\varepsilon^3 \gg \delta^2 \Rightarrow \varepsilon \gg \delta^{\frac{1}{3}} \tag{5.4.12}$$

so the parameter ε is much greater than (of lesser order than) δ, which describes the magnitude of the perturbation to the tearing mode flux function. However, this does not affect the subsequent results.

5.5 Magnetic field due to a tearing mode

As in the Finn and Kaw analysis, we take for a background magnetic field a tanh profile current sheet:

$$\underline{B}_0 = B_{0x}\,\hat{\underline{x}} + B_{0y}\,\hat{\underline{y}} \qquad (5.5.1)$$

$$B_{0x} = 0 \qquad B_{0y} = \tanh x$$

with the associated flux function

$$\psi_0(x,y) = \log_e \cosh x \qquad (5.5.2)$$

where

$$\underline{B}_0 = \underline{\nabla}\psi \times \hat{\underline{z}} \qquad (5.5.3)$$

$\underline{B}_1$ and hence ψ_1 are calculated by a linearisation and Fourier decomposition in y and t of the incompressible 2-D MHD equations. The technique is described in Priest (1985). Away from the neutral line, diffusion is negligible compared to advection so that terms of order S^{-1} may be set to zero. This gives an equation for $\bar{B}_{1x}$ in the external region

$$-\frac{\bar{B}_{1x} B_{0y}''}{B_{0y}} + (\bar{B}_{1x}'' - k^2 \bar{B}_{1x}) = 0 \qquad (5.5.4)$$

where k is the wavenumber of a Fourier mode in the y-direction, and

$$B_{1x}(x,y) = \bar{B}_{1x}(x)\cos ky \qquad (5.5.5)$$

With B_{0y} as above

$$\bar{B}_{1x}'' - k^2 \bar{B}_{1x} = -2\,\bar{B}_{1x}\,\mathrm{sech}^2 x \qquad (5.5.6)$$

where

$$\bar{B}_{1x} \in C^\infty(\{x,y) : x, y \in \mathbb{R}, x \neq 0\}) \qquad (5.5.7)$$

and

$$\bar{B}_{1x} \to 0 \quad \text{as} \quad x \to \pm\infty \qquad (5.5.8)$$

(5.5.6) has the form of a Schrodinger equation, and may be solved in terms of hypergeometric functions to give the result (normalised to unity at x = 0):

$$\psi_1(x,y) = e^{-k|x|}\left(1 + \frac{\tanh|x|}{k}\right)\cos ky \qquad (5.5.9)$$

where k may take any non-negative value. This is the solution consistent with the external boundary conditions, together with $\psi_1 = 0$ at $y = \pm\frac{\pi}{2k}$

Note that the x-derivative of ψ_1 has a discontinuity at x = 0. If the solution were continuous over all x, the spectrum of eigenvalues of (5.5.6) would reduce to a single value (k = 1) and the equation would have solution

$$\psi_1(x,y) = \operatorname{sech} x \cos y \qquad (5.5.10)$$

This is the first order flux function for the Finn and Kaw profile.

Given ψ_1 , we may examine the stability of this tearing mode. Define

$$\Delta' = \left. \frac{2\psi_1'}{\psi_1} \right|_{x=0} \qquad (5.5.11)$$

(Bateman, 1978). The mode will be unstable and will grow in time if Δ' > 0. In this case

$$\Delta' = 2\left(\frac{1}{k} - k\right) \qquad (5.5.12)$$

so we restrict the analysis below to k in the range (0,1].

We may now derive an expression for $J(\psi)$ up to first order. Since

$$J(\psi) = -\nabla^2\psi$$

$$J(\psi + \varepsilon\psi_1) = -\operatorname{sech}^2 x\,(1 - 2\varepsilon\psi_1)$$

$$= -e^{-2\psi_0}(1 - 2\varepsilon\psi_1)$$

The term in brackets on the right gives the first two terms of an exponential expansion. Since the form of J to higher orders is arbitrary, we may add in terms of order ε^2 and higher as long as consistency to the lowest two orders is retained:

$$J(\psi) = -\prod_{i=0}^{\infty} e^{-2\varepsilon^i \psi_i} \qquad (5.5.13)$$

$$= -e^{-2\psi} \qquad (5.5.14)$$

leading to the convenient expression

$$\frac{\partial J}{\partial \psi} = 2e^{-2\psi} \qquad (5.5.15)$$

This assumption is crucial to the form of ψ_2 and higher orders.

We may now quantify Hayashi's assertion that the current distribution for a tearing mode has the opposite phase to that of the Finn and Kaw profile. Figure (5.1) plots out the flux functions and current distributions to first order for both the Finn and Kaw and the tearing mode profiles. For the plots of magnetic field the X- and O-points coincide. However, a peak on one plot of J corresponds to a hollow on the other, and vice versa.

5.6 Boundary conditions

A suitable set of boundary conditions must be found for ψ and ζ to allow coalescence. We therefore impose that the wavelength of the perturbation ζ must be an integer multiple of the wavelength of ψ, and that ζ_i is zero at $y = \pm \frac{\pi}{2k}$. Note that with this condition ψ_i and ζ_i will not always have the same zeros. Suitable general expressions for ψ_i and ζ_i are

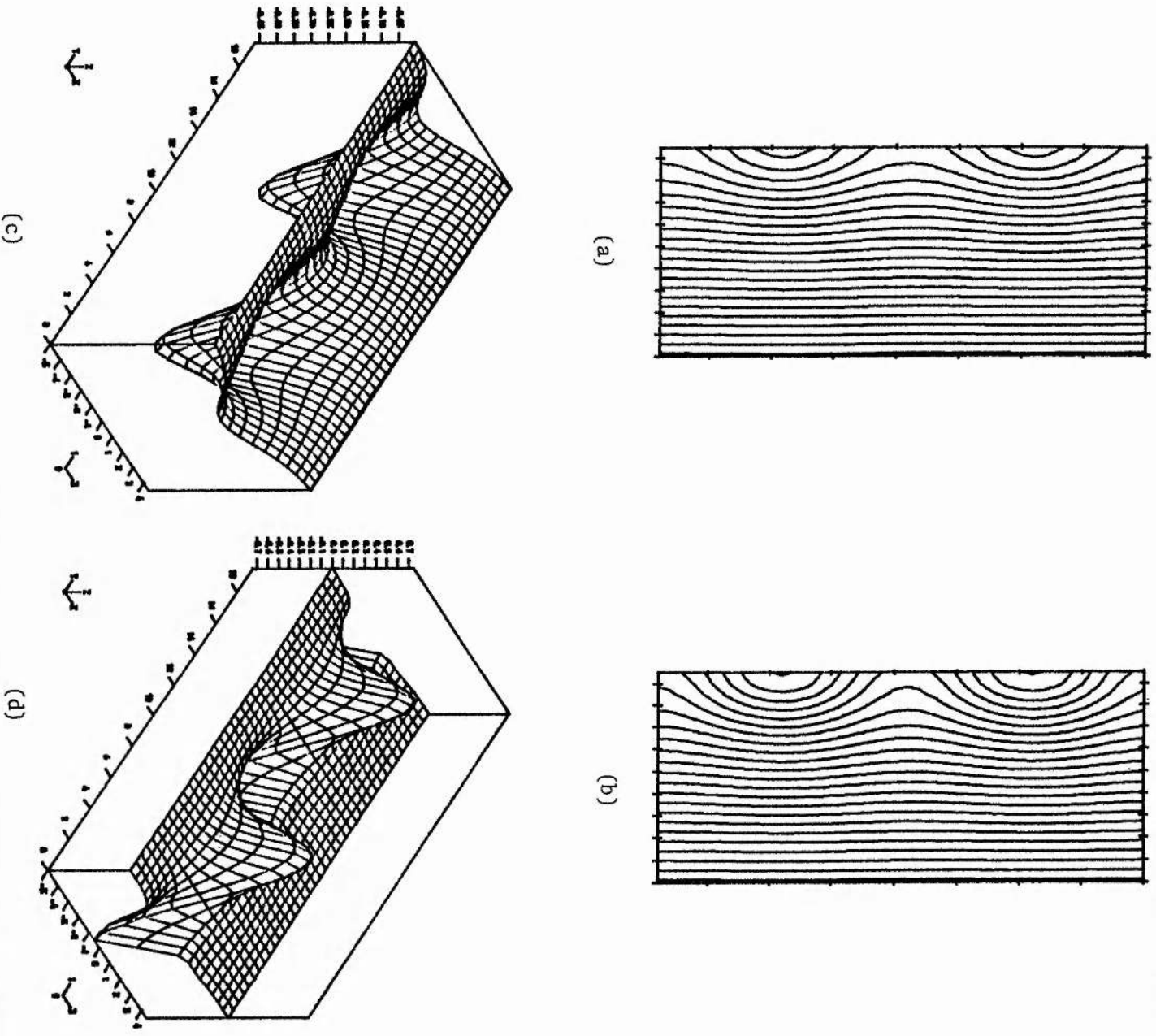


Figure 5.I Magnetic fields, current densities from the Finn and Kaw and tearing mode profiles

(a): Magnetic field lines from a Finn and Kaw island chain

(b): Magnetic field lines from a tearing mode island chain

(c): Current density (J_z) from a Finn and Kaw island chain

(d): Current density (J_z) from a tearing mode island chain

$$\xi_i = \sum_{j=0}^{N(i)} \alpha_{ij}(x) \cos jky \qquad (5.6.1)$$

$$\psi_i = \sum_{j=0}^{N(i)} \beta_{ij}(x) \cos njky \qquad (5.6.2)$$

where

$$\alpha_{ij}, \beta_{ij} \in C^\infty((x,y) : x, y \in \mathbb{R}, x \neq 0) \qquad (5.6.3)$$

n then gives the number of islands being forced together, and we are interested in the case $n \geq 2$.

It is now necessary to choose boundary conditions on α and β. We let

$$\alpha_{ij}(x), \beta_{ij}(x) \to 0 \text{ as } x \to \pm\infty \quad i,j \geq 1 \qquad (5.6.4)$$

Since the form of ψ must be given for each stability calculation, we normalise the coefficent functions β_{ij} $(i, j \geq 1)$ to unity at the origin:

$$|\beta_{ij}(0)| = 1 \qquad (5.6.5)$$

i.e. every harmonic, to every order, is normalised in this way. Note that this is an assumption, not a deduction, and that changing the normalisation will affect the stability results by altering the magnitude of the second term in the integral (5.9.14) below. We discuss this restriction in the conclusions to the chapter.

$$\beta_{00} = \log_e \cosh x \tag{5.6.6}$$

$$\beta_{10} = 0 \tag{5.6.7}$$

$$\beta_{11} = e^{-kk|x|}\left(1 - \frac{\tanh x}{kk}\right) \tag{5.6.8}$$

5.7 Ideal stability of a current sheet

The zero order Euler-Lagrange equation

$$-\nabla^2 \xi_0 - (\mu_0 + 2\,\mathrm{sech}^2 x)\,\xi_0 = 0 \tag{5.7.1}$$

describes both the zero order coalescence problem and an unperturbed current sheet. The equation may be separated by writing

$$\xi_0(x,y) = X(x)\,Y(y) \tag{5.7.2}$$

giving

$$\frac{Y''}{Y} + \mu_0 = -\frac{X''}{X} - 2\,\mathrm{sech}^2 x = \alpha \tag{5.7.3}$$

where α is a (real or imaginary) separation constant. The x equation is then

$$X'' + (\alpha + 2\,\mathrm{sech}^2 x)\,X = 0 \qquad (5.7.4)$$

which has general solution

$$X(x) = Ae^{-\sqrt{\alpha}|x|}\left(1+\frac{\tanh|x|}{\alpha}\right) + Be^{\sqrt{\alpha}|x|}\left(1-\frac{\tanh|x|}{\alpha}\right) \qquad (5.7.5)$$

Since $X \to 0$ as $x \to \infty$, $\alpha > 0$ for consistency with the boundary conditions (non-oscillatory decay as $x \to \infty$).

The y equation is

$$Y'' + (\mu_0 - \alpha)\,Y = 0 \qquad (5.7.6)$$

Define a wavenumber $k = (\mu_0 - \alpha)^{\frac{1}{2}}$. Since the boundary conditions on y require Y to be periodic, k must be real. This implies

$$k^2 = (\mu_0 - \alpha) \geq 0 \qquad (5.7.7)$$

and hence

$$\mu_0 \geq \alpha \geq 0 \qquad (5.7.8)$$

Such a current sheet is thus ideally stable to all perturbations. To proceed with a higher order analysis we let

$$\xi_0(x,y) \equiv 0 \qquad \mu_0 = 0 \qquad (5.6.9)$$

so that any negative contribution to μ from the higher order equations is retained.

The reader will observe that with the substitution (5.7.9) the two equations (5.4.6), (5.4.7) assume an apparent formal similarity. Despite this they are not the same equation and do not have the same solution, except for the particular eigenvalue we have chosen in (5.7.9). This is a partial differential equation with an eigenvalue term, and the spectrum of eigenvalues must be determined together with the respective eigenfunctions. $\mu_0 = 0$ is determined *a posteriori*, and this value is then substituted (along with the corresponding ξ_0) into the higher order Euler-Lagrange equations. For equation (5.4.6), the right hand side is known to be zero *a priori*; there is no eigenvalue to determine and hence one less term in the p.d.e. In fact, Fourier decomposition of (5.4.6) gives the tearing mode equation (5.5.6). However, all we require for further analysis is that $L(\xi_1) = 0$.

5.8 First order stability

We derive μ_1 by using a solvability condition (Nayfeh, 1984, p 358). The order ϵ^2 equation is

$$L(\xi_2) = \left.\frac{\partial J}{\partial \psi}\right|_1 \xi_1 + \mu_1 \xi_1 \qquad (5.8.1)$$

where the second and fourth terms in (5.4.8) are dropped. We multiply (5.8.1) by ξ_1 and integrate over the symmetric region D

$$\left\{(x,y) : -\infty < x < \infty , -\frac{\pi}{2k} \le y \le \frac{\pi}{2k}\right\}$$

containing n islands:

$$\int_D \xi_1 L(\xi_2)\, dS = \int_D \xi_1 \left[\frac{\partial J}{\partial \psi}\Big|_1 \xi_1 + \mu_1 \xi_1 \right] dS \qquad (5.8.2)$$

where dS = dx dy. The operator L is selfadjoint. The left hand side is zero on an application of Green's theorem because $L(\xi_1) = 0$ from (5.4.6) when $\xi_0 = 0$, $\mu_0 = 0$ are substituted. Hence

$$\int_D \frac{\partial J}{\partial \psi}\Big|_1 \xi_1^2\, dS = -\mu_1 \int_D \xi_1^2\, dS \qquad (5.8.3)$$

Since

$$\frac{\partial J}{\partial \psi}\Big|_1 = -4\psi_1^2 \operatorname{sech}^2 x \cos \pi k y \qquad (5.8.4)$$

and

$$\xi_1^2 = \bar{\xi}_1^2 \cos^2 \pi k y \qquad (5.8.5)$$

the integral of their product is an odd function. The integral of an odd function over a symmetric interval is zero. The integral on the right is non-zero by the normalisation constraint, so

$$\mu_1 = 0 \qquad (5.8.6)$$

5.9 Second order stability

We repeat the solvability condition analysis of Section 5.8 to find μ_2.

(5.4.9) now reduces to

$$L(\xi_3) = \frac{\partial \bar{J}}{\partial \psi}\Big|_1 \xi_2 + \frac{\partial \bar{J}}{\partial \psi}\Big|_2 \xi_1 + \mu_2 \xi_1 \qquad (5.9.1)$$

Multiplying through by ξ_1 and integrating over n islands leads to

$$\int_D \left[\frac{\partial \bar{J}}{\partial \psi}\Big|_1 \xi_2 + \frac{\partial \bar{J}}{\partial \psi}\Big|_2 \xi_1 \right] \xi_1 \, dS = -\mu_2 \int_D \xi_1^2 \, dS \qquad (5.9.2)$$

Expanding (5.5.14) to order ε^2 gives

$$\nabla^2 \psi_2 + 2\psi_2 \operatorname{sech}^2 x = \psi_1^2 \operatorname{sech}^2 x \qquad (5.9.3)$$

Since $\psi_1 = \bar{\psi}_1 \cos kky$, we may write

$$\psi_2 = \beta_{20}(x) + \beta_{22}(x) \cos 2kky \qquad (5.9.4)$$

Consider the first term in the integrand on the l.h.s.:

$$\int_D \frac{\partial \bar{J}}{\partial \psi}\Big|_1 \xi_1 \xi_2 \, dS \qquad (5.9.5)$$

ξ_2 may be found by solving

$$L(\xi_2) = \frac{\partial \bar{J}}{\partial \psi}\Big|_1 \xi_1 = \frac{\partial \bar{J}}{\partial \psi}\Big|_1 \bar{\xi}_1 \cos kky \cos ky \qquad (5.9.6)$$

$$= A(x)\cos(1+n)ky + B(x)\cos(1-n)ky \quad (5.9.7)$$

in which case the complementary function, found from

$$L(\xi_2) = 0 \quad (5.9.8)$$

is

$$\xi_2 = \bar{\xi}_2(x)\cos ky \quad (5.9.9)$$

and the particular integral is

$$\xi_2 = C(x)\cos(1+n)ky + D(x)\cos(1-n)ky \quad (5.9.10)$$

In both cases substitution back into (5.9.5) and evaluation of the y-component gives zero. Thus the integral reduces to

$$\int_D \frac{\partial J}{\partial \psi}\Big|_2 \xi_1^2\, dS = -\mu_2 \int_D \xi_1^2\, dS \quad (5.9.11)$$

From (5.4.15)

$$\frac{\partial J}{\partial \psi} = 2e^{-2\psi} = -2J(\psi) = 2\nabla^2\psi \quad (5.9.12)$$

giving

$$\int_D 2\xi_1^2 \nabla^2\psi_2\, dS = -\mu_2 \int_D \xi_1^2\, dS \quad (5.9.13)$$

We solve this equation numerically. From (5.9.3)

$$\int_D 2\zeta_1^2 \nabla^2 \psi_2 \, dS = \int_D 2\zeta_1^2 [\psi_1^2 - 2\psi_2] \operatorname{sech}^2 x \, dS \qquad (5.9.14)$$

$$= \int_D 2\zeta_1^2 [\psi_1^2 - 2(\beta_{20} + \beta_{22} \cos 2ky)] \operatorname{sech}^2 x \, dS \qquad (5.9.15)$$

The third term in the integrand gives zero contribution. The first term may be evaluated in terms of the known quantities ζ_1, ψ_1. It remains to find β_{20}. Substituting (5.9.4) into (5.9.3) and taking the components of the zero order harmonic gives

$$\frac{d^2\beta_{20}}{dx^2} + 2\beta_{20} \operatorname{sech}^2 x = \bar{\psi}_1^2 \operatorname{sech}^2 x \qquad (5.9.16)$$

where $\beta_{20} \to 0$ as $x \to \infty$. This ordinary differential equation is solved numerically by a Runge-Kutta scheme which integrates the equation from x = 30 to x = 0. The solution is then renormalised to unity at x = 0 for consistency with the boundary condition (5.6.5) and substituted into the integral (5.8.15) which is also evaluated numerically. Results for differing k are presented in figure (5.2).

It appears that a perturbation which forces together four or more islands will always cause coalescence, but for smaller island chains the lower wavelength islands will be stable. In particular, Hayashi's experiment of forcing together two k = 0.5 islands lies (just) within the region of stability. This is in agreement with his results.

In a real situation we would expect perturbations to an island chain to contain all harmonics rather than the discrete wavelengths considered here, so coalescence would always occur.

5.10 Conclusions

We have shown that islands resulting from a tearing mode instability are in almost all cases subject to a coalesence instability, but that the one relevant simulation performed to date by Hayashi (1981) lies within the region of stability. Given that his simulation lies very close to marginal stability on Figure 5.2, it appears that his boundary conditions are also having some stabilising effect. Instead of a proper freefloating boundary, he 'ties' all variables at this edge to their values at the start of the calculation. While the precise effect this will have on the stability properties of the experiment is unclear, it can hardly be anything other than stabilising.

The analysis we have presented has been mainly limited in applicability by the normalisation assumption (5.6.5). We have presented results that assume every harmonic, to every order, is normalised to unity. However, the validity of this assumption is not clear. Clarification would be helped by performing a second order expansion of the incompressible MHD equations (as well as the first order expansion assumed in the linearisation) and solving the resultant ODEs to find the second order flux function. In particular, this would then determine the ratio of the various β in (5.6.2) so that the order ϵ^2 harmonic could be normalised, and not just the individual components.

The other major limitation of this work is that the dynamics of the internal region have been neglected, and that the external solution has

been assumed to continue up to the line x = 0. We believe this is a valid approximation for a study of the ideal coalescence instability, where the driving force comes from the external region. A more detailed analysis involving the field profile of the inner region and a matching across the boundary layer should only give a slight modification to these results.

We suggest a few other avenues for further work. A generalisation of Hayashi's experiment to other wavelengths and perturbations will confirm or disprove the stability results shown on Figure 5.2, as long as a physically realistic 'free floating' boundary condition is implemented, which is difficult to do. A simpler configuration to model numerically is a double tearing mode, since only symmetry conditions need be assumed at the boundaries, and it may be possible to extend this analysis to determine the stability of such a field profile. Such an analysis would involve studying the role of perturbations perpendicular to the current sheet, which are also of interest in our simpler single tearing mode problem. One way to simplify the mathematics of this and related problems would be to find a representation of the magnetic islands in terms of fields generated by line currents along the z-axis.

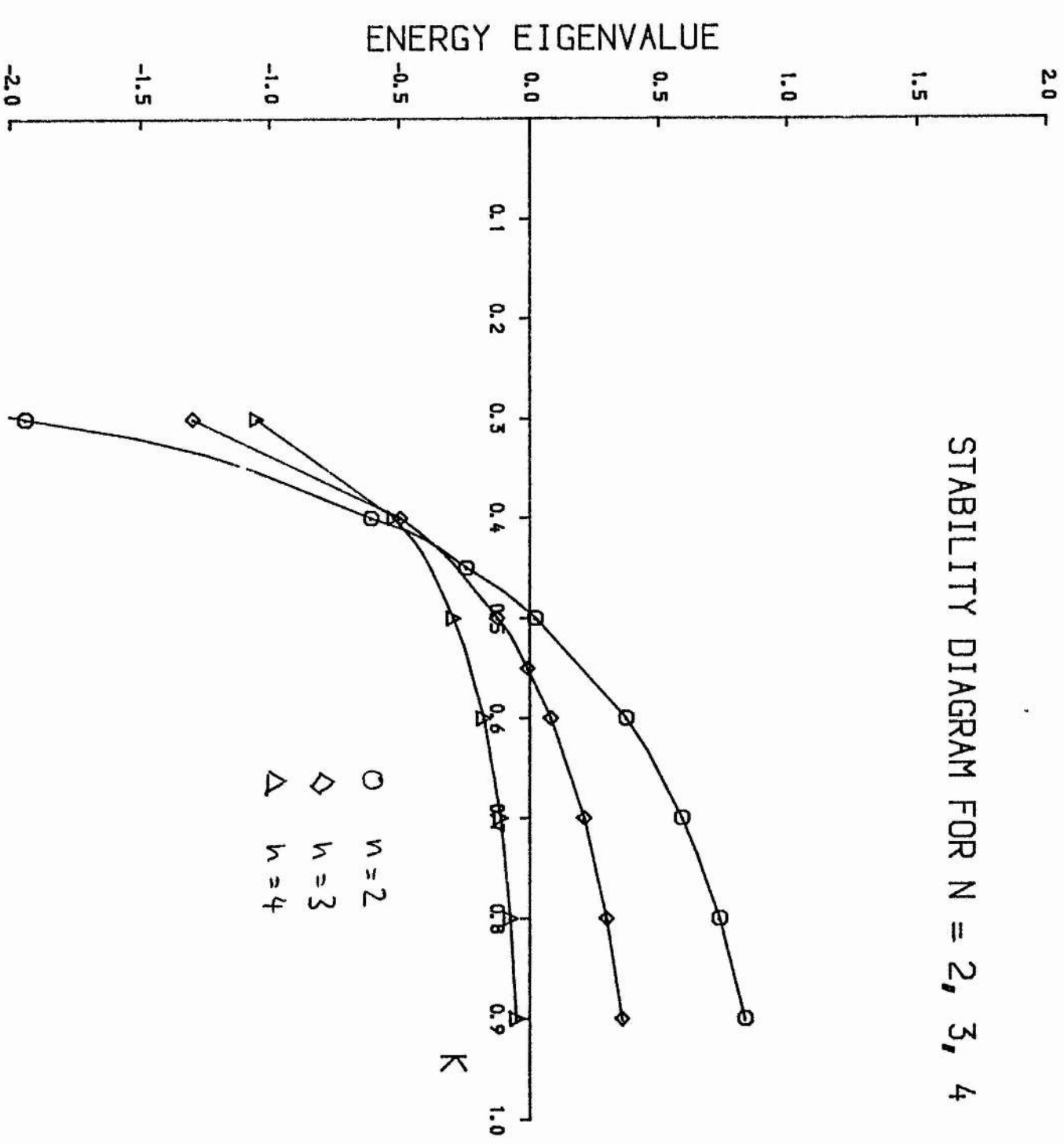


Figure 5.2 Value of the energy integral against wavenumber

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 Summary

In this thesis we have studied three problems: (i) The derivation of asymptotically steady-state solutions to the 2-D resistive MHD equations on an Eulerian mesh by use of a sophisticated finite difference code; (ii) Flow patterns arising from the linear and non-linear evolution of a tearing mode instability under different sets of boundary conditions, using the same code; (iii) The application of techniques from the calculus of variations to a study of ideal instability of chains of magnetic islands. In more detail:

(i) We have adapted a computer code that solves the six coupled partial differential equations of two-dimensional, compressible, resistive magnetohydrodynamics in Cartesian geometry, and used it for a systematic study of magnetic reconnection. The code employs the advection algorithm SHASTA which is particularly well suited for the treatment of shock waves. Such an algorithm will lead to greater accuracy in our solution, where we expect to find slow MHD shock waves. The resistive, or parabolic, part of the equations is solved by an implicit algorithm, so that the timestep required for each iteration has been limited only by the CFL condition. We have also avoided the problems of numerical dispersion and diffusion that arise when an explicit diffusion algorithm is used.

Previous work has shown the importance of the role of boundary conditions in such studies, and so we have taken particular care to ensure that their numerical implementation does not affect the accuracy

of the solutions. For the free-floating boundary we have chosen the best compromise between numerical accuracy and difficulty in implementation, while for the driven inflow boundary we have examined the role of compatability relations in the formation of boundary layers. We have also found, by experimentation, the best set of boundary conditions to impose at the inflow boundary that will initiate driven reconnection without causing numerical instability.

With a working code, we have studied the role of varying three parameters; the magnetic Reynolds number, the curvature of incoming field lines, and the velocity of the plasma as it enters the numerical box. This study has been limited by the resolution attainable with the computer power available at the time of the study, and by the very subtle interplay between the internally governed behaviour of a reconnecting system and the boundary conditions. However, our results indicate that the main parameter that governs the reconnction rate is the magnetic Reynolds number, and that the velocity field flow patterns may be dominated by the boundary conditions. For instance, compare Figures 6.1 and 6.2 with the diagrams shown in Chapter 4. The plots in this chapter show the steady-state reconnection solutions to the MHD equations with almost zero inflow speed but with high imposed field line curvature; those in Chapter 4 have effectively the same boundary conditions but with zero imposed field line curvature. The behaviour of the plasma flow patterns is completely different, an effect entirely due to the field line curvature.

(ii) Adapting the same code by imposing different boundary conditions, we have studied the temporal development of a tearing mode instability. We perturb a current sheet so that it tears, and impose a freefloating

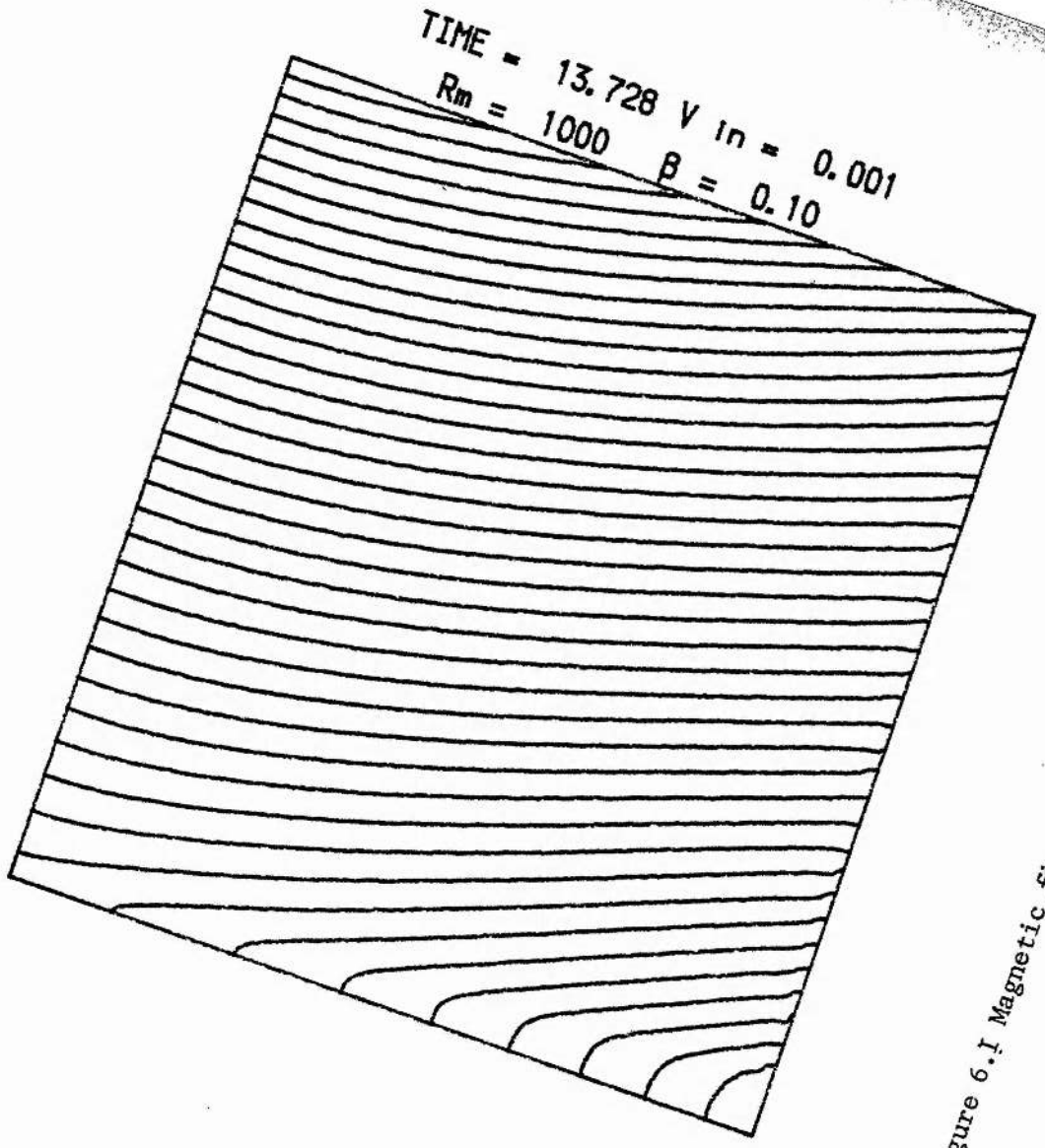


Figure 6.1 Magnetic field lines

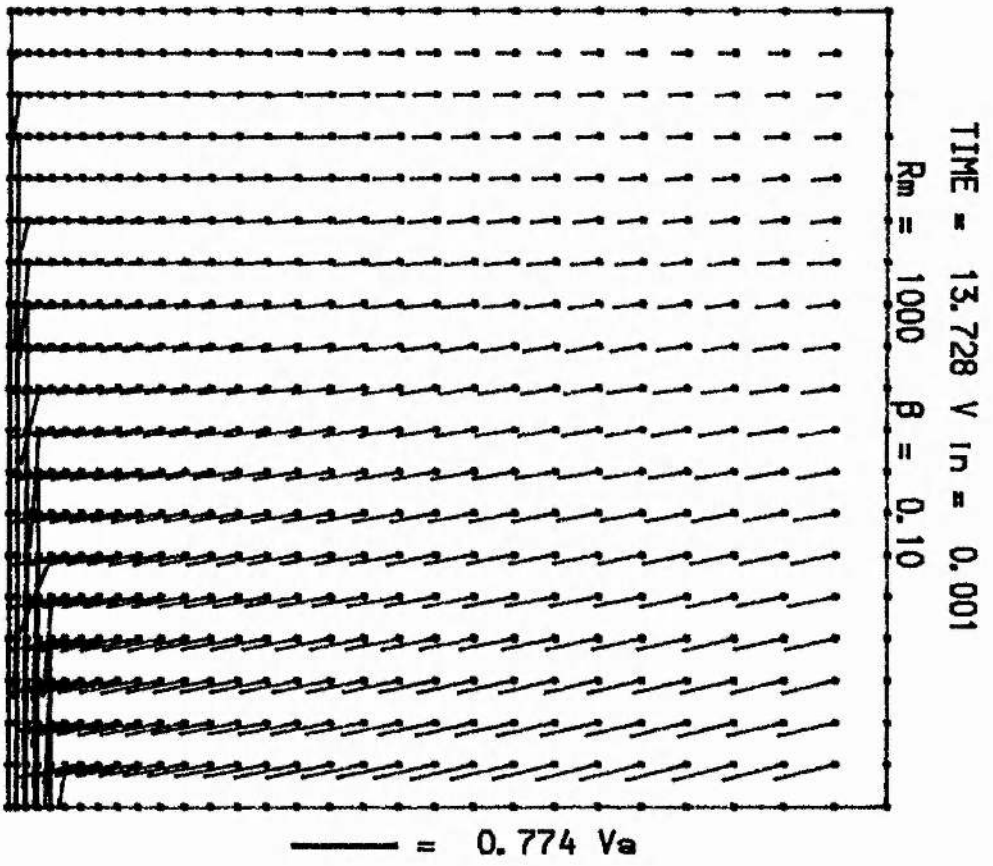


Figure 6.2 Plasma flow vectors

boundary parallel to the sheet which allows free entry and exit of plasma and magnetic flux. In the coordinate perpendicular to the current sheet we implement two different sets of boundary conditions; another free floating boundary, and an axis of symmetry which imposes a periodic symmetry on the solution.

We find that for the periodic case a tearing mode flow appears, with inflow at the X-point and outflow at the O-point. The resulting vortex is localised about the current sheet, with almost no activity occurring in the external region. By contrast, a jet forms over the freefloating boundary in the non-periodic simulation, but the flow patterns do not resemble those seen when the Petschek mechanism is operating. Again, little significant activity is seen in the external region.

We found difficulty in comparing our results to the standard analytical theory of tearing modes because of the low magnetic Reynolds numbers attainable. In particular, we were unable to study the linear development phase of the tearing mode since the high perturbation levels required to initiate formation of a magnetic island saturated the tearing mode within a few Alfven transit times. Also, Ohmic diffusion of the current sheet appeared to have an inhibitory effect on the growth rates, with the electric field at the X-point decreasing in time rather than increasing. We are confident that these problems could be solved with more computing power, enabling greater resolution at the current sheet and correspondingly higher magnetic Reynolds numbers.

For the analytical portion of this thesis we have studied the ideal magnetohydrodynamic instability of a chain of periodic magnetic islands which form in the linear growth phase of a tearing mode. We have demonstrated that the magnetic configuration most theorists have worked

with to date does not represent a tearing mode flux function, and we have calculated such a function by a linearisation of the MHD equations. After forming the Euler-Lagrange equation by a minimisation of the energy integral, we expand in powers of a small parameter which is proportional to the island width. The results have shown that such a chain of islands is stable to zeroth order and marginally stable to first order. Stability to second order depends on the wavenumber of the perturbation and of the islands thenselves, but we expect that for all cases of physical interest coalescence will occur.

6.2 Future work

It appears that the successful implementation of a driven inflow boundary for the full, hyperbolic-parabolic resistive MHD equations is a full-scale research problem in its own right; we are not aware of any published work presented to date in which even a one-dimensional simulation of this type has been successful. As a consequence there remains a slight degree of ambiguity about the cause of the boundary layer on the inflow boundary; is it due to overspecification of the inflow variables, or are the internal parameters within the plasma imposing reconnection rates and inflow speeds which are inconsistent with those imposed on the boundary? Clearly, further studies of this problem must address themselves to and eliminate the former cause, since physically realistic boundary conditions are of vital importance in all MHD fluid simulations. Given that the range of inflow parameters we may impose (such as inflow velocity) is limited, we may impose all the inflow parameters and reason that since only a minor 'glitch' occurs in the field variables on passing over the inflow boundary the

overspecification does not stop us attaining physically reasonable solutions beyond the glitch, even though they may not necessarily be the ones we would like.

Apart from the resolution of this problem, we suggest that finer resolution within the current sheet be implemented, which will enable us to take correspondingly higher magnetic Reynolds numbers so that the simulation will be applicable without reservation to astrophysical phenomena ($Rm > 10^{4}$). This should reveal a whole range of new phenomena, such as tearing within the current sheet, expulsion and coalescence of the resulting magnetic islands, and the resulting effect on the reconnection rate. Unfortunately, the computer power required for such a simulation appears to be beyond the capacity of any machine existing today.

By contrast, the work presented in Chapter 4 on tearing modes presents no new problems beyond the derivation of advection and diffusion algorithms that allow extreme variations in grid spacing. Eastwood and Arter's theory (1985) predicts that, with sufficient resolution inside the current sheet, the code should be stable under most initial conditions. If our code did not have the limitation of only an order of magnitude difference between gridspacing at inner and outer boundaries, we expect that many of the problems encountered would not have occurred. It would also be interesting to see the effect of changing the geometry, by (say) doubling the length scale in the x and z-directions.

The analytical work in Chapter 5 may be checked by setting up a numerical simulation of a chain of islands and imposing perturbations. This would not be difficult to do with the code used in Chapters 3 and

4. However, it is of crucial importance to set up a freefloating boundary parallel to the current sheet which does not affect the stability of the island chains. Further analytical work is necessary in three dimensions to examine the coupling of flux tubes formed by tearing modes with different helicities. Also, it would be of interest to study the effect of more general pertubations in two dimensions; what, for instance, is the role of a perturbation that acts perpendicular to the direction of the island chain? Such a study may be simplified by exhibiting a configuration of parallel current carrying wires out of the x-z plane that generate a tearing mode flux function, and then studying the stability of these wires to perturbations.

Appendix A

Derivation of the characteristic equations and compatability relations from the two-dimensional ideal magnetohydrodynamical equations

We expect the external region of our driven inflow simulation to be current-free, so that diffusive terms in the two-dimensional MHD equations may be neglected. In this case the equations reduce to

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v_x) + \frac{\partial}{\partial z}(\rho v_z) = 0 \tag{A.1}$$

$$\frac{\partial}{\partial t}(\rho v_x) + \frac{\partial}{\partial x}(\rho v_x v_x) + \frac{\partial}{\partial z}(\rho v_x v_z) = -\frac{\partial}{\partial x}\left(p + \frac{1}{2} B_z^2\right) + B_z \frac{\partial B_x}{\partial z} \tag{A.2}$$

$$\frac{\partial}{\partial t}(\rho v_z) + \frac{\partial}{\partial x}(\rho v_x v_z) + \frac{\partial}{\partial z}(\rho v_z v_z) = -\frac{\partial}{\partial z}\left(p + \frac{1}{2} B_x^2\right) + B_x \frac{\partial B_z}{\partial x} \tag{A.3}$$

$$\frac{\partial B_x}{\partial t} = \frac{\partial}{\partial y}(v_x B_z - v_z B_x) \tag{A.4}$$

$$\frac{\partial B_z}{\partial t} = \frac{\partial}{\partial x}(v_z B_x - v_x B_z) \tag{A.5}$$

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial x}(p v_x) + \frac{\partial}{\partial z}(p v_z) = -(\gamma - 1) p \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z}\right) \tag{A.6}$$

Consider the z-scan. In this case all variation in the x-direction vanishes, and the the equations become

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial z}(\rho v_z) = 0 \tag{A.7}$$

$$\frac{\partial}{\partial t}(\rho v_x) + \frac{\partial}{\partial z}(\rho v_x v_z) = B_z \frac{\partial B_x}{\partial z} \tag{A.8}$$

$$\frac{\partial}{\partial t}(\rho v_z) + \frac{\partial}{\partial z}(\rho v_z v_z) = -\frac{\partial}{\partial z}\left(p + \frac{1}{2} B_x^2\right) \tag{A.9}$$

$$\frac{\partial B_x}{\partial t} = \frac{\partial}{\partial z}(v_x B_z - v_z B_x) \tag{A.10}$$

$$\frac{\partial B_z}{\partial t} = 0 \tag{A.11}$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial z}(\rho v_z) = -(\gamma - 1) p \frac{\partial v_z}{\partial z} \tag{A.12}$$

From (A.11), B_z may be regarded as constant during the z-scan. Rewriting in non-conservative form, we have

$$\frac{\partial \rho}{\partial t} + v_z \frac{\partial \rho}{\partial z} + \rho \frac{\partial v_z}{\partial z} = 0 \tag{A.13}$$

$$\frac{\partial v_x}{\partial t} + v_z \frac{\partial v_x}{\partial z} = \frac{B_z}{\rho} \frac{\partial B_x}{\partial z} \tag{A.14}$$

$$\frac{\partial v_z}{\partial t} + v_z \frac{\partial v_z}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} - \frac{B_x}{\rho}\frac{\partial B_x}{\partial z} \qquad \text{(A.15)}$$

$$\frac{\partial B_x}{\partial t} = v_x \frac{\partial B_z}{\partial z} + B_z \frac{\partial v_x}{\partial z} - v_z \frac{\partial B_x}{\partial z} - B_x \frac{\partial v_z}{\partial z} \qquad \text{(A.16)}$$

$$\frac{\partial p}{\partial t} + v \frac{\partial p}{\partial z} = -\gamma p \frac{\partial v_z}{\partial z} \qquad \text{(A.17)}$$

However, it is more convenient for further analysis to work in terms of the entropy S. Equation (A.13) may be transformed using

$$S = \log_e p/\rho^\gamma \qquad \text{(A.18)}$$

to give

$$\frac{\partial S}{\partial t} + v_z \frac{\partial S}{\partial z} = 0 \qquad \text{(A.19)}$$

and so the matrix equation becomes

$$\frac{\partial}{\partial t}\begin{bmatrix} S \\ v_x \\ v_z \\ B_x \\ p \end{bmatrix} + \begin{bmatrix} v_z & 0 & 0 & 0 & 0 \\ 0 & v_z & 0 & -B_y/\rho & 0 \\ 0 & 0 & v_z & B_x/\rho & 1/\rho \\ 0 & -B_z & B_x & v_z & 0 \\ 0 & 0 & \gamma p & 0 & v_z \end{bmatrix} \frac{\partial}{\partial z}\begin{bmatrix} S \\ v_x \\ v_z \\ B_x \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -v_x \frac{\partial B_z}{\partial z} \\ 0 \end{bmatrix} = \underline{0} \qquad \text{(A.20)}$$

or in more compact notation,

$$\frac{\partial \underline{v}}{\partial t} + A \frac{\partial \underline{v}}{\partial z} + \underline{S} = 0 \qquad \text{(A.21)}$$

The eigenvalues of A are

$$\lambda = v_z \,,\; v_z \pm v_s \,,\; v_z \pm v_f \qquad \text{(A.21)}$$

where

$$v_s^2 = \tfrac{1}{2}\left[(a^2+b^2) - \{(a^2+b^2)^2 - 4a^2 v_a^2\}^{\frac{1}{2}}\right]$$

$$v_f^2 = \tfrac{1}{2}\left[(a^2+b^2) + \{(a^2+b^2)^2 - 4a^2 v_a^2\}^{\frac{1}{2}}\right]$$

$$a^2 = \frac{\gamma p}{\rho} \qquad b^2 = \frac{1}{\rho}\left(B_x^2 + B_t^2\right) \qquad v_a^2 = \frac{B_x^2}{\rho}$$

The characteristic equations follow by projecting this vector equation onto the eigenvectors:

$$\underline{\eta} \cdot \frac{\partial \underline{v}}{\partial t} + \underline{\eta} A \frac{\partial \underline{v}}{\partial z} + \underline{\eta} \cdot \underline{S} = 0 \qquad \text{(A.22)}$$

$$\Rightarrow \quad \underline{\eta} \cdot \left\{ \frac{\partial \underline{v}}{\partial t} + \lambda \frac{\partial \underline{v}}{\partial z} \right\} + \underline{\eta} \cdot \underline{S} = 0 \qquad \text{(A.23)}$$

so the five characteristic equations are

$$(1) \quad \frac{\partial S}{\partial t} + v \frac{\partial S}{\partial z} = 0 \qquad \text{(A.24)}$$

(i.e. entropy propagates at speed v)

$$(2)-(5) \quad \alpha B_x B_z D v_x - \alpha \rho (\alpha^2 - v_a^2) D v_z + \alpha^2 B_x D B_x - \alpha^2 B_x v_x \frac{\partial B_z}{\partial z} + (\alpha^2 - v_a^2) D p \qquad (A.25)$$

where

$$D \equiv \frac{\partial}{\partial t} + (v - \alpha) \frac{\partial}{\partial y} \qquad \alpha = \pm v_s, \pm v_f$$

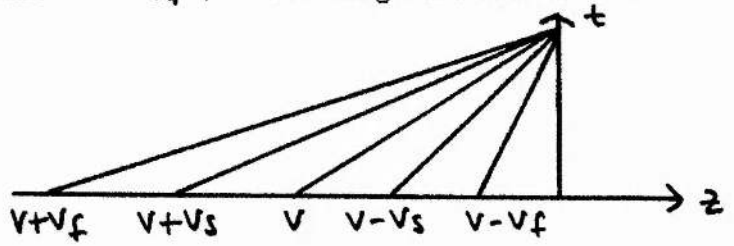
On using equation (A.18), (A.24) becomes

$$a^2 \frac{\partial \rho}{\partial t} - \frac{\partial p}{\partial t} = -a^2 v_z \frac{\partial \rho}{\partial z} + v_z \frac{\partial p}{\partial z} \qquad (A.26)$$

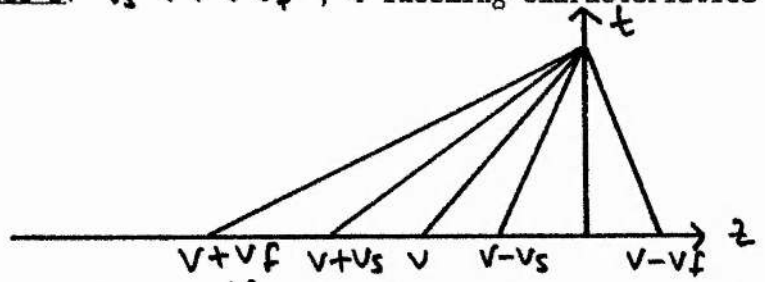
Consider a boundary where v > 0 imples inflow, v < 0 implies outflow. There are six cases to consider:

Inflow:

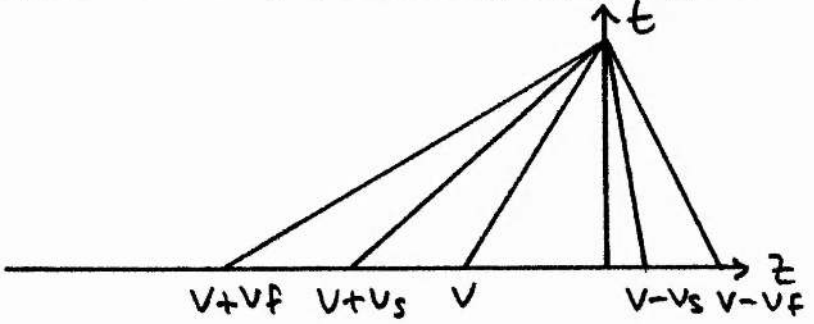
Case 1: v > v_f , 5 incoming characteristics



Case 2: v_s < v < v_f , 4 incoming characteristics

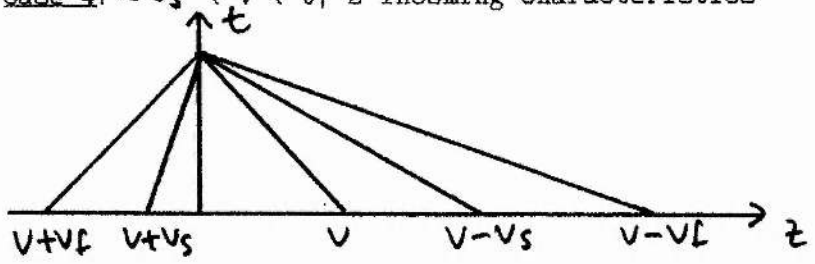


Case 3: 0 < v < v_s , 3 incoming characteristics

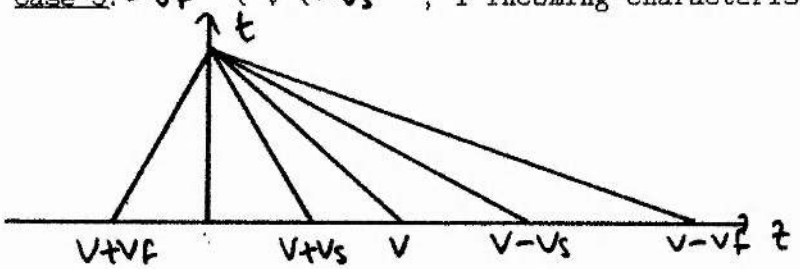


Outflow:

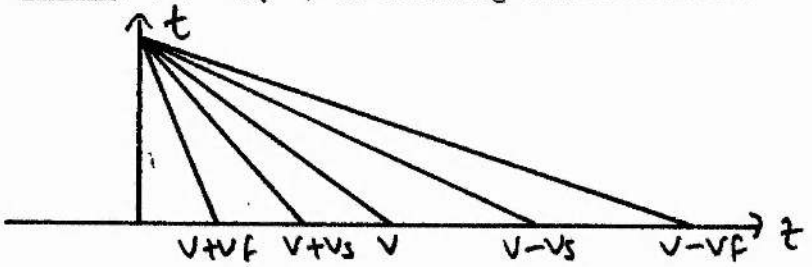
Case 4: $-V_S < v < 0$, 2 incoming characteristics



Case 5: $-V_f < v < -V_s$, 1 incoming characteristic



Case 6: $v < -V_f$, no incoming characteristics



For the z-scan B_z is fixed. If there is inflow, the value of the entropy S must be specified since S is then propagated away from the boundary. For an outflow S is found from

$$\frac{\partial S}{\partial t} + v \frac{\partial S}{\partial y} = 0 \qquad \text{(A.27)}$$

which will give ρ from equation (A.18). This leaves four variables to specify or derive (B_x, u, v, ρ):

Case		Number of fixed variables	
		S	(B_x, u, v, ρ)
1	(inflow)	1	4
2	(inflow)	1	3
3	(inflow)	1	2
4	(outflow)	0	2
5	(outflow)	0	1
6	(outflow)	0	0

In Case 1 all variables may be specified at an inflow boundary, so there is no need for a compatability relation. In Case 2 one compatability relation is required, and in case 3 two are required.

Appendix B

Listing of the computer code SHASTA

In Appendix B we give a complete listing of the Fortran code (named after the flux-corrected transport algorithm used in the advection routines) that produced the results in Chapters Three and Four. The main body of the program is the same for both cases and is given in Part 1. The implementation of boundary and initial conditions for the asymptotically steady state and the tearing mode problems are listed in Parts 2 and 3.

Part One: Advective and diffusive routines

```
      OPEN(UNIT=8,NAME='REDUCE.DAT',TYPE='NEW',FORM='UNFORMATTED')
      CALL ALFVEN
      CLOSE(UNIT=8)
      STOP
      END
      SUBROUTINE ALFVEN
C         MAIN PROGRAM ALFVEN 2 (40,70) IN  FORM
CDC   PROGRAM ALFVEN1(INPUT,OUTPUT,RESPUT,RESUIT,REDUCE,PLOT,DUMP,
CDC  + TAPE5=INPUT,TAPE6=OUTPUT,TAPE4=RESPUT,TAPE9=RESUIT,TAPE8=REDUCE,
CDC  + TAPE10=DUMP,TAPE99=PLOT)
C         USE OF TAPES: 4 AND 9 FOR CONTINUING COMPUTATION,
C                       8 FOR TREATMENT OF OUTPUT DATA REPRESENTATION,
C                      10 FOR DIAGNOSTIC PURPOSES.
C
C-----------------------------------------------------------------------
C       MEMORY STRUCTURE
C-----------------------------------------------------------------------
C         SWITCH CONTAINS 7 LOGICALS RULING THE PHYSICS
C         COMMUNICATED BETWEEN FYSICS,ADVECO,ADVECT,DIFFUX,DIFFUZ
C
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      COMMON/BFLDO/BZO(104),BXO(104),AO(104)
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
C        FYSVAR CONTAINS ALL INDEPENDENT PHYSICAL QUANTITIES
C        AND SOME COMBINATIONS WHICH ARE NECESSARY FOR EASY
C        REPRESENTATION OR RESTARTING.
C         LENGTH OF 8 ARRAYS (NX,NZ) AND 2 REALS.
C         IF NX,NZ LARGER THAN 40,70, FYSVAR MUST BE ADJUSTED
C         THROUGHOUT THE PROGRAM.
C
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
C
C         MEMORY COVERS FYSVAR FOR DATA DUMP.
C         LENGTH OF MEMORY IS TOTAL LENGTH OF FYSVAR,
C         OR 8*NX*NZ + 5. MEMDUM IS A DUMMY REAL FOR RESTART PURPOSES.
C         IN ISOTHERMAL CASES, MEMDUM CONTAINS THE TEMPERATURE.
C
      REAL MEMORY(22405)
      EQUIVALENCE (RHO(1,1),MEMORY(1))
C
C         METRIC CONTAINS COORDINATE SETS, LENGTHS AND SPACINGS.
C         D = DZ(MAX)/DX(MAX) IS A RATIO FACTOR FOR NON-SQUARE GRIDS.
C         D MUCH DIFFERENT FROM 1 WILL INFLUENCE THE TIME STEP.
C         METRIC COMMUNICATES WITH INITAL,ADVECO,TIMSTP,GEOMTR,ADVECT,
C         BOUNDY AND SUB-BOUNDY'S.
C         DUMMY LENGTH OF 99 IS USED FOR ALL ONE DIMENSIONAL
C         ARRAYS. IF NX OR NZ LARGER THAN 63, DUMMY LENGTH MUST
C         BE ADJUSTED THROUGHOUT THE PROGRAM.
C
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D,F
C
C         TIMING CONTAINS TIME STEP, MINIMUM AND
```

```
C          MAXIMUM SETTINGS AND COURANT NUMBERS FOR X-FLOW(AK1),
C          Z-FLOW(AK2),DIFFUSION(AK3). AK0 IS THE MAXIMUM ALOWED
C          CHANGE IN TIME STEP DURING ONE STEP (FOR REASONS OF
C          ACCIDENTS, TIME CONSUMPTION OR ACCURACY).
C          COMMUNICATES BETWEEN MAIN, TIMSTP AND SOLVE.
C
      COMMON/TIMING/DT,DTMIN,DTMAX,AK0,AK1,AK2,AK3
C
C          TESTER CONTAINS LTEST,LPRINT AND LINIT. IF LTEST IS TRUE,
C          CERTAIN S PRODUCE THEIR NAME ON FILE 10 WHEN
C          PASSED; THIS IS FOR DEBUGGING PURPOSES. IF LPRINT IS
C          FALSE, THE PRINT OUTPUT OF THE MAIN PROGRAM IS SUPPRESSED;
C          ONLY HEADINGS ARE GIVEN. IF LINIT IS FALSE, THE INITIAL STATE
C          IS NOT PRINTED. AFTER RESTARTING, THE INITIAL STATE IS NEVER
C          PRINTED.
C
      COMMON/TESTER/ LTEST,LPRINT,LINIT
C
C          CONSRC CONTAINS THE SOURCES FOR THE CONVECTION STEPS, AS
C          DEFINED IN ADVECO. ALL SOURCES ARE PUT ZERO AT THE BEGINNING.
C          THESE 6(EQUATIONS) TIMES 4(SOURCES) = 24 + 1(VELOCITIES IN
C          SCANNING DIRECTION) = 25 ARRAYS. BEYOND, FIVE BOUNDARY
C          CONDITIONS ARE SPECIFIED AS SET IN THE SHASTA ROUTINES:
C            ZBCX: CONDITION AT BOTTOM AND TOP OF X(VERTICAL) SCAN,
C            BCVRNZ: SYMMETRY CONDITION FOR X-FLOW MOMENTUM AT AXIS
C                  OF SYMMETRY,
C            BCVZ1Z: CONDITION AT OUTER BOUNDARY FOR Z-FLOW MOMENTUM.
C            BCBX1Z: SYMMETRY CONDITION FOR BX IN Z-SCAN AT OUTER
C                  BOUNDARY,
C            BCBXNZ: AS BCBX1Z BUT AT SYMMETRY AXIS.
C
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
C
C
C          DIFSRC CONTAINS SOURCES FOR DIFFUSION EQUATIONS AS DEFINED
C          IN DIFUCO. THESE ARE E1,E2 AND SE FOR THE THERMAL DIFFUSION,
C          B1 AND B2 FOR THE MAGNETIC DIFFUSION, AD,BD,CD,DD ARE THE
C          TRIDIAGONAL COEFFICIENTS AS DEFINED IN DIFUSE, B AND TE ARE
C          ARE ONE DIM. ARRAYS TO BE DIFFUSED, TENEW AND BNEW ARE
C          DIFFUSED TEMPERATURE     AND MAGNETIC FIELD COMPONENT.
C          ETA,ZKAPPA, KAPPAB ARE DIFFUSION CONSTANTS; MLSWPZ DIRECTS
C          THE SCAN DIRECTION.
C
      COMMON/DIFSRC/B1   (104),B2   (104),E1   (104),E2   (104),SE(104),
     2              AD   (104),BD   (104),CD   (104),DD   (104),
     3              TENEW(104),TE   (104),BNEW (104),B    (104),
     4              ETA  (104),KAPPAB(104),ZKAPPA(104),MLSWPZ
C
C          COVER (LENGTH 25 X LENGTH OF DUMMY ARRAYS) COVERS THE
```

```
C          ARRAYS PRESENT IN CONSRC (CONVECTION SOURCES), TO BE PUT
C          ZERO AT START, TO PREVENT UNDEFINED SOURCES.
C
      DIMENSION COVER(2600)
      EQUIVALENCE (COVER(1), SGRO(1))
C
C
C          SYSCON CONTAINS THE SCALE LENGTH RSC, TO BE EXPRESSED IN
C          SOLAR RADII, THE PLASMA BETA (GAS PRESSURE/MAGNETIC PR.),
C          THE LOGICAL ANISOT WHICH DETERMINES WHETHER KAPPA, SIGMA
C          AND ETA ARE TIME AND/OR COORDINATE DEPENDENT, ZBDIF AND
C          ZKDIF, WHICH ARE NUMERICAL FACTORS GIVING THE RELATION
C          BETWEEN REAL AND NUMERICAL SIGMA AND KAPPA RESPECTIVELY,
C          AND WHICH ARE USED AS 1/ETA RESP. KAPPA IF ANISOT=.F.,
C          AND FINALLY CPLRAD, WHICH IS A COUPLING FACTOR FOR
C          RADIATIVE LOSS PROCESSES.
C          PSI IS THE RATIO OF GRAVITATIONAL OVER KINETIC ENERGY
C          OF THE SCALE VALUES, PSI= G*MSUN/RSUN*VSC(1MK)
C
      COMMON/SYSCON/ RSC,ANISOT,ZBDIF,ZKDIF,CPLRAD,PSI
      COMMON/PARAM/WIDTH,RAP,DELON
C
C          MODO CONTAINS 1 INTEGER CONSTANT WHICH IS OPTIONALLY FOR
C          A PROGRAM MODUS CHOICE. MODO RULES INITAL, BOUNDY AND
C          ADVECT FOR INITIAL AND BOUNDARY CONDITIONS. SEE INITAL.
C          THE LOGICAL ARRAY SWTERM CONTAINS SWITCHES, READ IN FYSICS,
C          WHICH CAUSE INCLUSION OF DIV(G), S.GRAD(H) AND Q IN THE
C          SIX BASIC EQUATIONS. IF SWTERM(N,M) IS TRUE, THE TERM WITH
C          THIS LABEL IS NOT INCLUDED (NOT INCLUDED). N REFERS TO THE
C          EQUATION NUMBER, M TO THE SOURCE NUMBER, SEE FYSICS.
C
C
      COMMON/MODO  / MODUS,SWTERM(6,3)
C
C          BRAIN IS USED TO STORE SOME INITIAL VALUES TO BE USED FOR
C          TIME DEPENDENT BOUNDARY CONDITIONS OR FOR KEEPING A BOUNDARY
C          STRICTLY CONSTANT.
C          THE USER CAN USE THE ARRAYS FOR ARBITRARY ARRAYS TO BE
C          KEPT IN MEMORY.
C
      COMMON /BRAIN/ ZNB(104),BXB(104)
C
C          COMMON ISOTHM COMMUNICATES TEMPERATURE IN ISOTHERMAL APPLICA-
C          TIONS. TEMP MUST, IF SWTEMP=.F., BE DEFINED IN INITAL.
C
      COMMON/ISOTHM/ TEMP
      REAL KAPPAB,MEMDUM
C
C          RSTART RULES CONTINUING PROGRAM. IF FALSE, RUN IS FIRST
C          RUN AND INITAL IS USED. IF TRUE, FINAL STAGE OF PREVIOUS
C          RUN IS USED AS INITIAL CONDITION FOR NEXT RUN. THIS ASKS
C          FILE MANIPULATIONS WITH RESPUT AND RESUIT FILES.
C
C          RESTARTING IS USED TO PREVENT TOO LONG RUNS OR TOO MUCH
C          OUTPUT PER RUN. ONE IS ENABLED TO CHECK THE RESULTS DURING
```

```
C          VERY LONG RUNS AND IF NECESSARY TO CUT OFF COMPUTATION.
C
      LOGICAL LT,LF,RSTART,ANISOT,LTEST,LPRINT,LINIT,SWTERM,MLSWPZ
      LOGICAL SWADIF
C
C          ARRAY DIMENSIONS UP TO 11, 13 ARE POSSIBLE. TO CHANGE
C          DIMENSIONS, ONE MUST REDEFINE COMMON/FYSVAR/,MEMORY IN
C          MAIN,INITAL,BOUNDY,SUB-BOUNDY'S,ADVECO,ADVECT,MHDX,MHDZ,
C          TIMSTP,DIFFUZ,DIFFUX,DIFUCO,DIFUSE,SOLVE,DCOEFF,GEOMTR.
C
C-----------------------------------------------------------------------
C          DATA INPUT, CONSTANTS, DEFINITIONS
C-----------------------------------------------------------------------
      LF = .FALSE.
      LT = .TRUE.
      SWADIF = .TRUE.
C
C          INITIATION DATA
C
C-----------------------------------------------------------------------
C          DEFINE NATURE OF PHYSICAL PROBLEM
C
      CALL FYSICS
C
C          INPUT: 7 LOGICALS (FORMAT L1) IN FIRST COLUMN, F OR T
C                 SWGRAV, T FOR INCLUSION OF GRAVITATION
C                 SWTEMP, T FOR ALLOWING T-GRADIENTS
C                 SWRADN, T FOR INCLUSION OF RADIATIVE PROCESSES
C                 SW2DIM, T FOR 2-DIM CALCULATIONS
C                 SWMHDB, T FOR MAGNETIC FIELD (MHD) INCLUSION
C                 SWTDIF, T FOR T-DIFFUSION IF SWTEMP = .T.
C                 SWBDIF, T FOR B-DISSIPATION IF SWMHDB = .T.
C
C                 SWTERM(6,3) (6 TIMES 3 LOGICALS RULING THE INCLUSION
C                 (.F.) OR SKIPPING (.T.) OF A SOURCE TERM, SEE FYSICS).
C                 MODUS (FORMAT I1) CHOSING ONE OF TWO INITIAL STATE
C                 EXAMPLES.
C                 ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ (FORMAT 5F4.0),
C                 BEING THE BOUNDARY CONDITIONS INCLUDED BY SHASTA.
C
C                 THE NEXT INPUT CARD IS TO BE FILLED WITH AT MOST 80
C                 CHARACTERS TEXT DESCRIBING THE RUN (OPTIONAL).
C
C
C-----------------------------------------------------------------------
C          INPUT: NX,NZ (FORMAT I2,I4) FOR ARRAY DIMENSIONS
C          I MAXIMUM FOR NX AND NZ = 13
C
      READ(5,8000) NX,NZ
      IF(.NOT.SW2DIM) NZ=1
      READ(5,1008) D
      READ(5,1008) F
      CALL GEOMTR
C-----------------------------------------------------------------------
C
```

```
C          SUPPLY AUXILIARY DATA
C
C          MAGNETIC REYNOLDS NUMBER
C          PLASMA BETA
C          INFLOW VELOCITY
C          (FORMAT 3F6.4)
C
      READ(5,1001) REYN,BETA,VINP
      PRINT 1001,REYN,BETA,VINP
C
C          DEFINITION OF COURANT NUMBERS FOR FAST-MODE SPEED,
C          X-FLOW SPEED AND Z-FLOW SPEED RESP. SEE TIMSTP FOR
C          DESCRIPTION.
C
      AK0 = 05.0
      AK1 = 0.20
      AK2 = 0.20
      AK3 = 0.20
      READ(5,1002) LTEST,LPRINT,LINIT
C          FORMAT(3L1). IF T, DEBUGGING DATA ARE WRITTEN TO TAPE10,
C          AND TABLES ARE PRINTED AFTER JS TIME STEPS AND AT THE INITIAL
C          POINT.
C
      READ(5,1003) RSTART
C          (FORMAT L1). IF .F., INITIAL CONDITIONS SET BY INITAL
C                       IF .T., RESTART FROM PERMANENT FILE DATA
C
      READ(5,1005) ANISOT
C          (FORMAT L1)
C          IF ANISOT = .T. CALL TO DCOEFF IS NECESSARY EACH TIME STEP
C          TO EVALUATE DIFFUSION TRANSPORT COEFFICIENTS ETA AND KAPPA
C          IF ANISOT = .F. VALUES OF SIGMA AND KAPPA ARE DEFINED
C          BY 1/ZBDIF AND ZKDIF
C
      READ(5,1006) RSC,ZBDIF,ZKDIF,CPLRAD
      READ(5,1006) WIDTH,BETA,RAP,DELON
C          RSC IS THE COMPUTATIONAL LENGTH SCALE, EXPRESSED IN SOLAR
C          RADII (OTHER STANDARDS POSSIBLE, REQUIRE REDEFINITION OF
C          DIFFUSION COEFFICIENTS CONSTANTS). BETA IS (IN MODUS 2)
C          THE PLASMA BETA AT THE SHEET BOUNDARY IN THE INITIAL STATE.
C          ZBDIF AND ZKDIF ARE ISOTROPIC DEVIATIONS FROM THE CLASSICAL
C          DIFFUSION COEFFICIENTS SIGMA AND KAPPA.
C          CPLRAD IS A CONSTANT FOR COUPLING THE RADIATIVE LOSSES
C          ACCORDING TO THE COX-TUCKER FUNCTION. CPLRAD.LT.1.0 WILL
C          SLOW DOWN THE RADIATIVE PROCESSES.
C
      READ(5,1007) JP,JS
C          (FORMAT I2,I4)
C          JP CYCLES ARE MADE PRODUCING OUTPUT
C          JS TIME STEPS ARE MADE BETWEEN OUTPUT STAGES
C
C
C          INITIAL STATE
C          READ DATA FROM RESPUT, IF RUN IS RESTARTED
C
```

```
      IF(RSTART) GOTO 20
C-----------------------------------------------------------------------
   10 CONTINUE
C
C          DEFINE INITIAL CONDITIONS
C
      CALL INITAL
C          INPUT: MODE (FORMAT I1)
C          SEE  INITAL
C          IF RESTART, THE INPUT CARD IS READ AND STORED IN DUMMY, SUCH
C          THAT THE INPUT RECORD FOR INITIAL AND RESTARTED RUNS IS
C          EQUAL EXCEPT FOR THE RESTART PARAMETER.
C
      GOTO 30
C-----------------------------------------------------------------------
   20 CONTINUE
C
C          RESTARTED RUN
C
C          RESTARTED DATA ARE READ FROM RESPUT TO MEMORY
C          MEMORY COVERS THE FYSVAR ARRAY
C
      REWIND 4
      READ(4) MEMORY
      IF(.NOT.SWTEMP) TEMP = MEMDUM
      DO 13 IRS = I,NX
      AO(IRS) = A(IRS,NZ)
      BXO(IRS) = BX(IRS,NZ)
   13 BZO(IRS) = BZ(IRS,NZ)
      PSI = 5.317
  183 CONTINUE
      TSHEET = TIME/WIDTH
      WRITE(6,2001) TIME


C
C-----------------------------------------------------------------------
C
   30 CONTINUE
      DO 35 IB=1,NZ
      ZNB(IB) =RHO(1,IB)
      BXB(IB) = BX(1,IB)
   35 CONTINUE
      IF(ANISOT) GOTO 50
C-----------------------------------------------------------------------
   40 CONTINUE
      NWA = MAXO(NX,NZ)
      DO 45 JKE=1,NWA
      ZKAPPA(JKE) = ZKDIF
      ETA(JKE) = 1.0/ZBDIF
   45 CONTINUE
C-----------------------------------------------------------------------
   50 CONTINUE
      DO 55 J = 1,NZ
      DO 55 I = 1,NX
```

```
      ES(I,J) = 0.0
   55 CONTINUE
C
C          MAGNETIC AND/OR HEAT SOURCES MUST BE SPECIFIED
C          FOR BS AND ES. THE ASSIGNMENT STATEMENT CAN BE REPLACED
C          BY CALLS TO SPECIFIC SUBROUTINES, SEE MAIN.272
C
C
C          RESET SOURCE TERMS TO ZERO.
C          RESETTING IS NOT ESSENTIAL
C          THE NUMBER OF TERMS IS 25 X 99.
C
      DO 56 KCOVER = 1,3750
   56 COVER(KCOVER) = 0.0
C
C          GIVE LAYOUT OF INITIAL STATE IF NOT RESTARTED
C
      IF(RSTART) GOTO 60
C
      WRITE(8) MEMORY
      IF(.NOT.LINIT) GOTO 60
      LINEP = NZ
      IF(.NOT.SW2DIM)LINEP = 1
      CALL PRINT(X,RHO ,NX,LINEP,NX,4H RHO)
      CALL PRINT(X,RHOVX,13,LINEP,NX,4H RVX)
      IF(SW2DIM)CALL PRINT(X,RHOVZ,NX,LINEP,NX,4H RVZ)
      IF(SW2DIM)CALL PRINT(X,BX  ,NX,LINEP,NX,4H BX )
      CALL PRINT(X,BZ  ,NX,LINEP,NX,4H BZ )
      CALL PRINT(X,U   ,NX,LINEP,NX,4H U  )
   60 CONTINUE
      CALL VTMSTP
C
C-----------------------------------------------------------------------
C          COMPUTATIONAL LOOPS
C          JP OUTPUT GIVING CYCLES ARE MADE EACH CONSISTING OF
C          JS TIME STEPS. JS MAY VARY IN TIME
C-----------------------------------------------------------------------
C
      DO 400 JJP = 1,JP
      DO 300 JJS = 1,JS
C
C          DIFFUSION STEPS ARE MADE IF EITHER SWBDIF OR SWTDIF = .T.
C
      IF (SWRADN) CALL RAD(NX,NZ)
      CALL AFROMB
      IF(.NOT.SWBDIF.AND..NOT.SWTDIF) GOTO 102
       CALL DIFFUX
       CALL BOUNDY(5,0)
       IF(SW2DIM) CALL DIFFUZ
       CALL BOUNDY(5,0)
  102 CONTINUE
      CALL BFROMA
C          CONVECTIVE STEPS METHOD OF VECTORISATION WITH SHASTA (BORIS,
C          1972 ,NRL-REPORT, SHAS2D)
C
```

```
C     GO TO 110
      CALL MHDX(DT)
      CALL BOUNDY(1,0)
      IF(.NOT.SW2DIM) GOTO 110
      CALL MHDZ(DT)
      CALL BOUNDY(1,0)
  110 CONTINUE
      TIME = TIME + DT
      CALL VTMSTP
      DO 973 TR = 1,NX
      OUO(TR) = U(TR,NZ)
973   ORHOO(TR) = RHO(TR,NZ)
  300 CONTINUE
C
      OPEN(12,FILE = 'EFIELD',STATUS = 'OLD',ACCESS = 'APPEND')
      CJ1 = (BX(2,3)-BX(2,1))/(Z(3)-Z(1))
      CJ2 = (BZ(3,2)-BZ(1,2))/(X(3)-X(1))
      ELECTRIC = (CJ1-CJ2)/REYN
      WRITE(12,1009)TIME,ELECTRIC
      CLOSE(11)
C         STORE DATA FOR RESTART. ONE RECORD IS KEPT AND CATALOGED AFTER
C         CORRECT FINISH OR EXIT.
C
      REWIND 9
      IF(.NOT.SWTEMP) MEMDUM = TEMP
      WRITE(9) MEMORY
C
C         STORE DATA FOR REDUCTION
C
      WRITE(8) MEMORY
      WRITE(6,2002) TIME
      IF(.NOT.LPRINT) GOTO 390
      LINEP = NZ
      IF(.NOT.SW2DIM)LINEP = 1
      CALL PRINT(X,RHO ,NX,LINEP,NX,4H RHO)
      CALL PRINT(X,RHOVX,NX,LINEP,NX,4H RVX)
      IF(SW2DIM)CALL PRINT(X,RHOVZ,NX,LINEP,NX,4H RVZ)
      IF(SW2DIM)CALL PRINT(X,BX  ,NX,LINEP,NX,4H BX )
      CALL PRINT(X,BZ  ,NX,LINEP,NX,4H BZ )
      IF(.NOT.SWTEMP) GOTO 390
      CALL PRINT(X,U   ,NX,LINEP,NX,4H U  )
C
C         CHECK TIME STEP
C
  390 CONTINUE
  400 CONTINUE
  500 CONTINUE
      WRITE(6,2003)
 1001 FORMAT(3F15.5)
 1002 FORMAT(3L1)
 1003 FORMAT(L1)
 1004 FORMAT(I1)
 1005 FORMAT(L1)
 1006 FORMAT(5E10.3)
 1007 FORMAT(I2,I4)
```

```
 1008 FORMAT(E10.3)
 1009 FORMAT(F9.5,F15.5)
 8000 FORMAT(I4,I4)
 2001 FORMAT(28H0 START TIME OF THIS RUN IS  ,F6.3)
 2002 FORMAT(31H0 OUTPUT AT ALFVEN SCALE TIME  ,F8.5//)
 2003 FORMAT(15H0       ALFVEN1,/,
     +       57H PROGRAM "ALFVEN" BASED ON CASTOR CODE (J.P.CHRISTIANSEN
     +),/,   56H USING PHOENICAL SHASTA (BORIS AND BOOK, NRL,WASHINGTON.
     + ,/,   46H VERSION U,1977. WEBER (UTRECHT OBSERVATORY),
     + ,/,   54H BORIS AND GARDNER (NAVAL RESEARCH LAB WASHINGTON DC).)
 2004 FORMAT(30H  PARTICLE   X-COORD   Z-COORD)
 2005 FORMAT (I10,2F10.3)
      RETURN
      E N D
      SUBROUTINE FYSICS
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      COMMON/MODO  /MODUS,SWTERM(6,3)
      LOGICAL SWTERM
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
      DIMENSION TEXT(20)
C
C          DETERMINATION OF PHYSICAL EFFECTS TO BE INCLUDED
C
  100 CONTINUE
      READ(5,1) SWGRAV
      READ(5,1) SWTEMP
      READ(5,1) SW2DIM
      READ(5,1) SWMHDB
      READ(5,1) SWRADN
      READ(5,1) SWTDIF
      IF(.NOT.SWTEMP) SWTDIF = .FALSE.
      READ (5,1) SWBDIF
      IF(.NOT.SWMHDB) SWBDIF = .FALSE.
      WRITE(6,9990)
      IF(.NOT.SWGRAV) WRITE (6,9991)
      IF(.NOT.SWTEMP) WRITE (6,9992)
      IF(.NOT.SWRADN) WRITE (6,9993)
      IF(     SW2DIM) WRITE (6,9994)
      IF(     SWMHDB) WRITE (6,9995)
      IF(.NOT.SWTDIF.AND..NOT.SWBDIF) WRITE(6,9996)
      IF(SWTDIF.AND..NOT.SWBDIF) WRITE(6,9997)
      IF(SWBDIF.AND..NOT.SWTDIF) WRITE (6,9998)
C     WRITE(6,9999)
C
C          DEFINITION OF TERMS TO BE INCLUDED IN EQUATIONS.
C          SIX SETS OF THREE PARAMETERS ARE READ, CORRESPONDING
C          TO THE SIX EQUATIONS TO BE INTEGRATED:
C              CONTINUITY EQUATION (1)
```

```
C             EQUATION OF MOTION (2)
C             MAGNETIC FIELD EQUATION (CONVECTIVE PART) (2)
C             ENERGY EQUATION (1)
C
C         THE THREE PARAMETERS FOR EACH EQUATION CORRESPOND TO
C         INCLUSION OF THE THREE TERMS AT RIGHT OF THE BASIC EQUATION
C             DF/DT = DIV(G) + S.GRAD(H) + Q
C
  200 CONTINUE
C         TERMS FOR CONTINUITY EQUATION
      READ(5,2)(SWTERM(1,I),I=1,3)
C         TERMS FOR EQUATION OF MOTION IN X-DIRECTION
      READ(5,2)(SWTERM(2,I),I=1,3)
C         TERMS FOR EQUATION OF MOTION IN Z-DIRECTION
      READ(5,2)(SWTERM(3,I),I=1,3)
C         TERMS FOR EQUATION OF FIELD  IN X-DIRECTION
      READ(5,2)(SWTERM(4,I),I=1,3)
C         TERMS FOR EQUATION OF FIELD  IN Z-DIRECTION
      READ(5,2)(SWTERM(5,I),I=1,3)
C         TERMS FOR ENERGY EQUATION
      READ(5,2)(SWTERM(6,I),I=1,3)
C
C
C         SELECTION OF MODUS FOR INITIAL CONDITIONS
C         (TO BE CHOSEN FROM 2 EXAMPLES)
C
  300 CONTINUE
      READ(5,3) MODUS
C     WRITE(6,9988) MODUS
C
C         READING OF BOUNDARY CONDITIONS FIXED IN SHASTA,
C         IN X-DIRECTION:                    ZBCX
C         IN Z-DIRECTION, AT KZ= 1: RHOVZ = BCVZ1Z
C                                   BX    = BCBX1Z
C                              NZ: RHOVX = BCVRNZ
C                                  BX    = BCBXNZ
C
      READ(5,9987) ZBCX, BCVRNZ, BCVZ1Z, BCBX1Z,BCBXNZ
C         READING OF AN EXPLAINING TEXT OF MAXIMUM 80 CHARACTERS
      READ(5,4)(TEXT(IK),IK=1,20)
      WRITE(6,9989)(TEXT(IK),IK=1,20)
    1 FORMAT(L1)
    2 FORMAT(3L1)
    3 FORMAT(I1)
    4 FORMAT(20A4)
 9987 FORMAT(5F4.0)
C9988 FORMAT(47H MODUS OF INITIAL CONDITIONS SELECTED: MODUS = ,I1,//)
 9989 FORMAT(1H ,20A4,//)
 9990 FORMAT(41HOPHYSICAL PROPERTIES OF SYSTEM: ALFPHYS  ,//)
 9991 FORMAT(16H NO GRAVITATION  )
 9992 FORMAT(16H ISOTHERMAL      )
 9993 FORMAT(19H NO RADIATIVE LOSS )
 9994 FORMAT(28H TWO-DIMENSIONAL CALCULATION )
 9995 FORMAT(21H MAGNETOHYDRODYNAMICS )
 9996 FORMAT(15H DIFFUSION FREE )
```

```
 9997 FORMAT(22H INFINITE CONDUCTIVITY )
 9998 FORMAT(21H NO THERMAL DIFFUSION )
 9999 FORMAT(16H          FYSICS)
      RETURN
      E N D
      SUBROUTINE GEOMTR
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D,F
      COMMON/MODO  /MODUS,SWTERM(6,3)
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      COMMON/PARAM/WIDTH,BETA,RAP,DELON
      REAL TOP,BOTTOM
      LOGICAL  SWTERM
      DZ = 1.0
987   FORMAT(E10.3)
988   FORMAT(//,2X,'WIDTH OF CURRENT SHEET =',E10.3)
      PRINT 988,D
986   FORMAT(//,2X,'GRID SPACING PARAMETER =',E10.3)
      PRINT 986,F
      DO 101 IX = 1,NX
  101 X(IX) = (FLOAT(IX-2))/(FLOAT(NX-2))
      IF(.NOT.SW2DIM) GOTO 103
      DO 102 IZ = 1,NZ
      TOP = FLOAT(IZ-2)*(2.0+(F*FLOAT(IZ-3)))
      BOTTOM = FLOAT(NZ-2)*(2.0+(F*FLOAT(NZ-3)))
      Z(IZ) = TOP/BOTTOM
      PRINT 987,Z(IZ)
102   CONTINUE
      CALL SSETBB(Z,Z,NZ,1,NZ)
  103 CALL SSETYY(X,X,NX,1,NX)
C     WRITE(6,9999)
 9999 FORMAT(16H          GEOMTR )
      RETURN
      E N D
      SUBROUTINE MHDX(DT)
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
      COMMON/METRIC/X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/MHDRZI/RHOI(104),RHOVXI(104),RHOVZI(104),BXI(104),BZI(104),
     2              UII   (104),
     3              RHOH(104),RHOVXH(104),RHOVZH(104),BXH(104),BZH(104),
     4              UHH   (104)
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      REAL MEMDUM
```

```
      LOGICAL LTEST,LPRINT,LINIT,MLSWPZ
C
C          CONVECTION IN X-DIRECTION.
C          TAKES PLACE IN TWO STEPS; PREDICTOR-CORRECTOR METHOD:
C          AFTER HALF STEP SOURCES G,S,H,Q ARE DETERMINED WHICH
C          THEN ARE TIME-CENTERED
C          BOUNDARY CONDITIONS ARE IMPOSED AFTER DT/2.
C
      MLSWPZ = .FALSE.
      DO 103 JZ=1,NZ
CU    DO 102 K=1,NX
CU    IF(.NOT.SWMHDB) BXH(K) = BX(K,JZ)
CU    IF(.NOT.SWMHDB) BZH(K) = BZ(K,JZ)
CU    IF(.NOT.SWTEMP) UHH(K) = U(K,JZ)
CU102 CONTINUE
      CALL ADVECO(RHO(1,JZ),RHOVX(1,JZ),RHOVZ(1,JZ),U (1,JZ),BX(1,JZ),
     +            BZ(1,JZ), NX, MLSWPZ,JZ)
      CALL ADVECT(RHO(1,JZ),RHOVX(1,JZ),RHOVZ(1,JZ),U (1,JZ),BX(1,JZ),
     + BZ(1,JZ),
     +  RHOH, RHOVXH,  RHOVZH,  UHH,  BXH,  BZH  ,
     +         NX, 0.5*DT, MLSWPZ)
      CALL BOUNDY(3,JZ)
C
C          BX NOT EVOLVED IN X-SCAN
C
C          SOURCES AT HALF TIME, CONVECTION FROM TIME ZERO TO
C          FULL TIME STEP. SECOND ORDER ACCURACY.
C
      CALL ADVECO(RHOH,RHOVXH,RHOVZH,UHH,BX(1,JZ),BZH,NX,MLSWPZ,JZ)
      CALL ADVECT(RHO(1,JZ),RHOVX(1,JZ),RHOVZ(1,JZ),U (1,JZ),BX(1,JZ),
     + BZ(1,JZ),
     +           RHO(1,JZ),RHOVX(1,JZ),RHOVZ(1,JZ),U (1,JZ),BX(1,JZ),
     + BZ(1,JZ),
     +        NX, DT,  MLSWPZ)
  103 CONTINUE
      IF(LTEST) WRITE(10,9999)
 9999 FORMAT(16H            MHDX )
      RETURN
      END
      SUBROUTINE MHDZ(DT)
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/MHDRZI/RHOI(104),RHOVXI(104),RHOVZI(104),BXI(104),BZI(104),
     2              UII   (104),
     3              RHOH(104),RHOVXH(104),RHOVZH(104),BXH(104),BZH(104),
     4              UHH   (104)
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL        SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      REAL MEMDUM
      LOGICAL LTEST,LPRINT,LINIT,MLSWPZ
C
```

```
C          CONVECTION IN Z-DIRECTION. SEE MHDX.
C          MHDZ IS NOT USED IN ONE-DIMENSIONAL APPLICATIONS.
C
C          FOR COLUMN TRANSPORT, INTERMEDIATE ARRAYS ARE FILLED.
C
      MLSWPZ = .TRUE.
      DO 108 JR=1,NX
      DO 101 JZ=1,NZ
      RHOI(JZ) = RHO(JR,JZ)
      RHOVXI(JZ) = RHOVX(JR,JZ)
      RHOVZI(JZ) = RHOVZ(JR,JZ)
      BXI(JZ) = BX(JR,JZ)
      BZI(JZ) = BZ(JR,JZ)
      UII   (JZ) = U (JR,JZ)
  101 CONTINUE
      CALL ADVECO(RHOI,RHOVXI,RHOVZI,UII,BXI,BZI,NZ,MLSWPZ,JR)
C
C          TO HALF STEP (PREDICTOR).
C
      CALL ADVECT(RHOI, RHOVXI, RHOVZI, UII, BXI, BZI,
     +            RHOH, RHOVXH, RHOVZH, UHH, BXH, BZH,
     +            NZ, 0.5*DT, MLSWPZ)
      CALL BOUNDY(4,JR)
C
C          BZ NOT EVOLVED IN Z-SCAN
C
CU    DO 100 K=1,NZ
CU    IF(.NOT.SWMHDB) BXH(K) = BX(JR,K)
CU    IF(.NOT.SWMHDB) BZH(K) = BZ(JR,K)
CU    IF(.NOT.SWTEMP) UHH(K) = U(JR,K)
CU100 CONTINUE
      CALL ADVECO(RHOH,RHOVXH,RHOVZH,UHH,BXH,BZI,NZ,MLSWPZ,JR)
C
C          TO FULL STEP (CORRECTOR).
C
      CALL ADVECT(RHOI, RHOVXI, RHOVZI, UII, BXI, BZI,
     +            RHOH, RHOVXH, RHOVZH, UHH, BXH, BZH,
     +            NZ, DT    , MLSWPZ)
      DO 102 JZ=1,NZ
      RHO(JR,JZ) = RHOH(JZ)
      RHOVX(JR,JZ) = RHOVXH(JZ)
      RHOVZ(JR,JZ) = RHOVZH(JZ)
  102  CONTINUE
      IF(.NOT.SWMHDB) GOTO 104
      DO 103 JZ=1,NZ
  103 BX(JR,JZ) = BXH(JZ)
  104 CONTINUE
  106 IF(.NOT.SWTEMP) GOTO 108
      DO 107 JZ=1,NZ
  107 U(JR,JZ) = UHH(JZ)
  108 CONTINUE
      IF(LTEST) WRITE(10,9999)
 9999 FORMAT(16H            MHDZ )
      RETURN
      END
```

```
      SUBROUTINE ADVECT(PRHOLD,PVROLD,PVZOLD,PEOLD,PBXOLD,PBZOLD,
     +                  PRHNEW,PVRNEW,PVZNEW,PENEW,PBXNEW,PBZNEW,
     +                  K,PDT,LSWEPZ)
      DIMENSION
     + PRHOLD(K),PVROLD(K),PVZOLD(K),PEOLD(K),PBXOLD(K),PBZOLD(K),
     + PRHNEW(K),PVRNEW(K),PVZNEW(K),PENEW(K),PBXNEW(K),PBZNEW(K)
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SWMHDB,SW2DIM
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
      COMMON/MODO  /MODUS,SWTERM(6,3)
      COMMON/TESTER/LTEST,LPRINT,LINIT
      COMMON/ISOTHM/ TEMP
      LOGICAL LT,LF,LSWEPZ,SWTERM,LTEST,LPRINT,LINIT
      LF = .FALSE.
      LT = .TRUE.
      IF(LSWEPZ) GOTO 200
C
C         ADVECT PERFORMS THE CALLS TO THE SHASTA ROUTINES FOR THE
C         REAL CONVECTION IN X- AND Z-DIRECTION. CONVECTION OF
C         NON-PRESENT QUANTITIES IS SKIPPED.
C
C         IN X-DIRECTION ARE SCANNED: RHO,RHOVX,RHOVZ,P,BZ. THE
C         BOUNDARY AT BOTTOM AND TOP IS THE SAME, TO BE CHOSEN
C         BETWEEN FREE FLOATING, PERIODIC AND REFLECTING.
C
C
C         BOUNDARY CONDITIONS FOR SHASTA.
C
C         MODUS = 1 : PERIODIC.
C
      ABSZBC = ABS(-1.0)
      CALL SSETVY(X,VELOCH,K,1,K,PDT)
      CALL SHASTY(PRHOLD,PRHNEW,
     +     SWTERM(1,1),SGRO,SWTERM(1,2),SSRO,SHRO,SWTERM(1,3),SQRO,
     +     K,1,K,ABSZBC)
      CALL SHASTY(PVROLD,PVRNEW,
     +     SWTERM(2,1),SGVR,SWTERM(2,2),SSVR,SHVR,SWTERM(2,3),SQVR,
     +     K,1,K,-1.0)
      IF(SW2DIM)
     +CALL SHASTY(PVZOLD,PVZNEW,
     +     SWTERM(3,1),SGVZ,SWTERM(3,2),SSVZ,SHVZ,SWTERM(3,3),SQVZ,
     +     K,1,K,ABSZBC)
      IF(SWTEMP.OR.SWBDIF)
     +CALL SHASTY(PEOLD ,PENEW ,
     +     SWTERM(6,1),SGEQ,SWTERM(6,2),SSEQ,SHEQ,SWTERM(6,3),SQEQ,
     +     K,1,K,ABSZBC)
      IF(.NOT.SWMHDB) GOTO 110
      CALL SHASTY(PBZOLD,PBZNEW,
```

```
     +      SWTERM(5,1),SGBZ,SWTERM(5,2),SSBZ,SHBZ,SWTERM(5,3),SQBZ,
     +      K,1,K,-1.0)
  110 CONTINUE
      IF(SWTEMP) GOTO 105
      DO 104 I=1,NX
C     PENEW(I) = 1.5*PRHNEW(I)*TEMP+0.5*(PVRNEW(I)**2+PVZNEW(I)**2)
C    +   + 0.5*(PBZNEW(I)**2+PBXOLD(I)**2)
      PENEW(I) = PRHNEW(I)*TEMP
  104 CONTINUE
  105 CONTINUE
      IF(LTEST) WRITE(10,9999)
      RETURN
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C         IN THE Z-DIRECTION ARE SCANNED: RHO,RHOVX,RHOVZ,P AND BX.
C         BOUNDARIES LEFT (AT KZ=NZ, NEAR FIELD AXIS) AND RIGHT
C         (AT KZ=1, OUTER BOUNDARY) CAN BE CHOSEN INDEPENDENTLY.
C         RESTRICTION: RHOVZ IS ALWAYS ANTISYMMETRIC AT KZ=NZ/NZ-2
C         BECAUSE OF THE KZ=NZ-1-LINE AS AN AXIS OF SYMMETRY.
C
C         NOT USED IN ONE-DIMENSIONAL COMPUTATIONS.
  200 CONTINUE
C
C         BOUNDARY CONDITIONS FOR SHASTA.
C
C         MODUS = 1: FREEFLOATING OUTER BOUNDARY.
C
C         MODUS = 2: REFLECTING OUTER BOUNDARY (TENSION REPRESENTATION)
C
      CALL SSETVB(Z,VELOCH,K,1,K,PDT)
      CALL SHASTB(PRHOLD,PRHNEW,
     +      SWTERM(1,1),SGRO,SWTERM(1,2),SSRO,SHRO,SWTERM(1,3),SQRO,
     +      K,1,K,1.0,1.0)
      CALL SHASTB(PVROLD,PVRNEW,
     +      SWTERM(2,1),SGVR,SWTERM(2,2),SSVR,SHVR,SWTERM(2,3),SQVR,
     +      K,1,K,1.0,-1.0)
      CALL SHASTB(PVZOLD,PVZNEW,
     +      SWTERM(3,1),SGVZ,SWTERM(3,2),SSVZ,SHVZ,SWTERM(3,3),SQVZ,
     +      K,1,K,-1.0,-1.0)
      IF(SWTEMP)
     +CALL SHASTB(PEOLD ,PENEW ,
     +      SWTERM(6,1),SGEQ,SWTERM(6,2),SSEQ,SHEQ,SWTERM(6,3),SQEQ,
     +      K,1,K,1.0,1.0)
      IF(.NOT.SWMHDB) GOTO 210
      CALL SHASTB(PBXOLD,PBXNEW,
     +      SWTERM(4,1),SGBR,SWTERM(4,2),SSBR,SHBR,SWTERM(4,3),SQBR,
     +      K,1,K,-1.0,-1.0)
  210 CONTINUE
      IF(SWTEMP) GOTO 205
      DO 204 J=1,NZ
C 204 PENEW(J)=1.5*PRHNEW(J)*TEMP + 0.5*(PVRNEW(J)**2+PVZNEW(J)**2) +
C    +      0.5*(PBZOLD(J)**2+PBXNEW(J)**2)
  204 PENEW(J) = PRHNEW(J)*TEMP
  205 CONTINUE
      IF(LTEST) WRITE(10,9999)
```

```
 9999 FORMAT(16H          ADVECT )
      RETURN
      E N D
      SUBROUTINE DIFFUX
      COMMON/DIFSRC/B1   (104),B2   (104),E1   (104),E2   (104),SE(104),
     2              AD   (104),BD   (104),CD   (104),DD   (104),
     3              TENEW(104),TE   (104),BNEW (104),B    (104),
     4              ETA  (104),KAPPAB(104),ZKAPPA(104),MLSWPZ
      COMMON/FYSVAR/RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2              BX   (40,70),BZ   (40,70),U    (40,70),
     3              ES   (40,70),A    (40,70),TIME ,MEMDUM,
     4              REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/BCAD/SWIMPL
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      REAL KAPPAB,MEMDUM
      LOGICAL MLSWPZ,LTEST,LPRINT,LINIT,SWADIF,SWIMPL
C
C         DIFFUX SOLVES DIFFUSION EQUATION IN X-DIRECTION.
C         DIFFUSION EQUATION: PF/PT = DIV (COEFF . GRAD F) + SF
C         WITH P=PARTIAL DERIVATIVE, DIV AND GRAD ARE APPROPRIATE
C         SPACE DERIVATIVES AND SF IS A NON-INFINITESIMAL SOURCE.
C
C         IN SEQUENCED ORDER DIFFUX DIFFUSES T,BX AND BZ.
C
      MLSWPZ = .FALSE.
      SWADIF = .TRUE.
C
C         SET DIFFUSION SOURCES AND COEFFICIENTS AND DIFFUSE IN TE.
C
      DO 122 JZ=1,NZ
      CALL DIFUCO(JZ,NX )
      IF(SWTDIF) CALL DIFUSE(JZ,1,NX,DX,1)
C
C         STORE DIFFUSED VALUE AND TAKE A LINE OF THE Z-FIELD.
C
      DO 118 I1 = 1,NX
      IF(SWTDIF) U(I1,JZ) = TENEW(I1)*RHO(I1,JZ)
CU?   IF(SWTDIF) U(I1,JZ) = 1.5*TENEW(I1)*RHO(I1,JZ) +
CU    + 0.5*(BX(I1,JZ)**2+BZ(I1,JZ)**2) +
CU    + 0.5*(RHOVX(I1,JZ)**2+RHOVZ(I1,JZ)/RHO(I1,JZ))
  118 B(I1) = BZ (I1,JZ)
C         SWITCH FOR A DIFFUSION
      IF(SWADIF) GO TO 124
      IF(.NOT.SWBDIF) GOTO 122
      CALL DIFUSE(JZ,1,NX,DX,2)
C
C         STORE DIFFUSED BZ-VALUE AND TAKE A LINE OF THE X-FIELD.
C
      DO 119 I2 = 1,NX
      BZ (I2,JZ) = BNEW(I2)
  119 B(I2) = BX (I2,JZ)
      IF(.NOT.SW2DIM) GOTO 122
```

```
C
C         DIFFUSE IN BX.
C
      MLSWPZ = .TRUE.
      CALL DIFUSE(JZ,1,NX,DX,2)
      MLSWPZ = .FALSE.
      DO 120 I3 = 1,NX
C
C         STORE DIFFUSED VALUE OF BX.
C
  120 BX (I3,JZ) = BNEW(I3)
      GO TO 122
C         A DIFFUSION IN X
  124 CONTINUE
      SWIMPL = .TRUE.
      DO 126 I4 = 1,NX
  126 B(I4) = A(I4,JZ)
      CALL DIFUSE (JZ,1,NX,DX,2)
C         STORE DIFFUSED A - VALUE
      DO 128 I4 = 1,NX
  128 A(I4,JZ) = BNEW(I4)
  122 CONTINUE
      IF(LTEST) WRITE(10,9999)
 9999 FORMAT(16H           DIFFUX )
      RETURN
      END
      SUBROUTINE DIFFUZ
      COMMON/DIFSRC/B1   (104),B2   (104),E1   (104),E2   (104),SE(104),
     2              AD   (104),BD   (104),CD   (104),DD   (104),
     3              TENEW(104),TE   (104),BNEW (104),B    (104),
     4              ETA  (104),KAPPAB(104),ZKAPPA(104),MLSWPZ
      COMMON/FYSVAR/RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2              BX   (40,70),BZ   (40,70),U    (40,70),
     3              ES   (40,70),A    (40,70),TIME ,MEMDUM,
     4              REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      COMMON/BCAD/SWIMPL
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      REAL KAPPAB,MEMDUM
      LOGICAL MLSWPZ,LTEST,LPRINT,LINIT,SWIMPL,SWADIF
C
C         DIFFUSION IN Z-DIRECTION, SEE DIFFUX.
C         SKIPPED IN ONE-DIMENSIONAL COMPUTATIONS.
C
      MLSWPZ = .TRUE.
      SWADIF = .TRUE.
C         1.1  DIFFUSE TE AND B IN Z-DIRECTION
      DO 122 JR=1,NX
      CALL DIFUCO(JR,NZ )
      IF(SWTDIF) CALL DIFUSE(JR,1,NZ,DZ,1)
      DO 118 I1 = 1,NZ
      IF(SWTDIF) U(JR,I1) = TENEW(I1)*RHO(JR,I1)
CU?   IF(SWTDIF) U(JR,I1) = 1.5*TENEW(I1)*RHO(JR,I1) +
```

```
CU    + 0.5*(BX(JR,I1)**2+BZ(JR,I1)**2) +
CU    + 0.5*(RHOVX(JR,I1)**2+RHOVZ(JR,I1)/RHO(JR,I1))
  118 B(I1) = BX (JR,I1)
C         SWITCH FOR A DIFFUSION
      IF(SWADIF) GO TO 124
      IF(.NOT.SWBDIF) GOTO 122
      SWIMPL = .FALSE.
      CALL DIFUSE(JR,1,NZ,DZ,2)
      DO 119 I2 = 1,NZ
      BX(JR,I2) = BNEW(I2)
  119 B(I2) = BZ (JR,I2)
      CALL DIFUSE(JR,1,NZ,DZ,2)
      DO 120 I3 = 1,NZ
  120 BZ (JR,I3) = BNEW(I3)
      SWIMPL = .FALSE.
      GO TO 122
C         A DIFFUSION IN Z
  124 CONTINUE
      SWIMPL = .TRUE.
      DO 126 I4 = 1,NZ
  126 B(I4) = A(JR,I4)
      CALL DIFUSE(JR,1,NZ,DZ,2)
C         STORE DIFFUSED A - VALUE
      DO 128 I4 = 1,NZ
  128 A(JR,I4) = BNEW(I4)
  122 CONTINUE
      IF(LTEST) WRITE(10,9999)
 9999 FORMAT(16H          DIFFUZ )
      RETURN
      END
      SUBROUTINE DIFUCO(KNOW,KNXZ)
      COMMON/DIFSRC/B1   (104),B2   (104),E1   (104),E2   (104),SE(104),
     2              AD   (104),BD   (104),CD   (104),DD   (104),
     3              TENEW(104),TE   (104),BNEW (104),B    (104),
     4              ETA  (104),KAPPAB(104),ZKAPPA(104),MLSWPZ
      COMMON/FYSVAR/RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2              BX   (40,70),BZ   (40,70),U    (40,70),
     3              ES   (40,70),A    (40,70),TIME ,MEMDUM,
     4              REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      REAL MEMDUM,KAPPAB
      LOGICAL ANISOT,MLSWPZ,LTEST,LPRINT,LINIT
      COMMON/SYSCON/ RSC,ANISOT,ZBDIF,ZKDIF,CPLRAD,PSI
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         SETTING OF COEFFICIENTS IN DIFFUSION EQUATION OF THE
C         GENERALISED FORM:
C         PF/PT = C1 DIV (C2 GRAD C3.F) + SF
C         WHERE C = E IF F = TEMPERATURE AND
C         C = B IF F = MAGNETIC FIELD. C1 DEPENDS ON POSITION,
C         C2 ON POSITION AND KAPPA OR SIGMA/ETA, C3 IS IDENTICAL TO
C         ONE. KAPPA AND ETA CAN BE TENSORS AND CAN BE INHOMOGE-
C         NEOUS AND ANISOTROPIC. HERE WE USE ISOTROPIC QUANTITIES.
```

```
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      SHARE = 1.0
      IF(SW2DIM) SHARE = 0.5
C
C         IN TWO-DIMENSIONAL CASES THE CONTRIBUTION OF THE SOURCE TERM
C         IS DIVIDED OVER BOTH SCANS.
C
C
C         THE SPECIFIC HEAT EQUALS 1/(GAMMA-1). FOR GAMMA = 5/3:
C         CV = 3/2. (GAS CONSTANT R IS RENORMALISED TO UNITY).
C
      CV = 1.5
      IF(MLSWPZ) GOTO 200
C         1.  RADIAL SWEEP(EVALUATE COEFFS FOR R DIFFUSION)
   40 CONTINUE
      DO  50 I=1,NX
C     IF(U(I,KNOW).LE.0.0) GOTO 300
      IF(U(I,KNOW).LE.0.0)U(I,KNOW) = 1.0E-20
      IF(RHO(I,KNOW).LE.0.0)RHO(I,KNOW) = 1.0E-20
      TE(I) = U(I,KNOW)/RHO(I,KNOW)
CU?   TE(I) =     (2./3.)*(U(I,KNOW)-0.5*(BX(I,KNOW)**2+BZ(I,KNOW)**2) -
CU    - 0.5*(RHOVX(I,KNOW)**2+RHOVZ(I,KNOW)**2)/RHO(I,KNOW))/RHO(I,KNOW)
      SE(I) = 0.0
   50 CONTINUE
C
C         DETERMINE COEFFICIENTS IN DIFFUSION EQUATION.
C
      IF(ANISOT) CALL DCOEFF(KNOW,NX)
C
C          OHMIC HEATING IS INCLUDED ONLY, IF SIGMA IS REDUCED SIGNIFI-
C          CANTLY.
C
C     IF(SWBDIF.AND.ZBDIF.LT.1E-4) CALL OHM(KNOW,MLSWPZ)
      ZNCV = CV*RHO(KNXZ,KNOW)
      E1(KNXZ) = 1.0/ZNCV
      E2 (KNXZ) = 0.0
      SE (KNXZ) = SHARE * (ES(KNXZ,KNOW) + SE(KNXZ))
      B1(KNXZ) = 1.0
      B2 (KNXZ) = 0.0
      KM1 = KNXZ - 1
      DO 104 JR=1,KM1
      ZNCV = CV*RHO(JR,KNOW)
      E1(JR) = 1.0/ZNCV
      E2(JR) =(ZKAPPA(JR) + ZKAPPA(JR+1)) * 0.5
      SE ( JR ) = SHARE * (ES( JR ,KNOW) + SE( JR ))
      B1(JR) = 1.0
      B2(JR) = 0.5 * (ETA(JR) + ETA(JR+1))
  104 CONTINUE
      IF(LTEST) WRITE(10,9997)
      RETURN
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         2.  AXIAL SWEEP(EVALUATE COEFFS FOR Z DIFFUSION)
  200 CONTINUE
```

```
  110 CONTINUE
      KCO=KNOW
      DO 150 I=1,NZ
C963   FORMAT('   ',E10.5)
C       PRINT 963,U(KNOW,I)
C     IF(U(KNOW,I).LT.0.0) GOTO 300
      IF(U(KNOW,I).LT.0.0)U(I,KNOW) = 1.0E-20
      IF(RHO(KNOW,I).EQ.0.0)RHO(KNOW,I) = 1.0E-15
      TE(I) = U(KNOW,I)/RHO(KNOW,I)
CU?   TE(I) =     (2./3.)*(U(KNOW,I)-0.5*(BX(KNOW,I)**2+BZ(KNOW,I)**2) -
CU    - 0.5*(RHOVX(KNOW,I)**2+RHOVZ(KNOW,I)**2)/RHO(KNOW,I))/RHO(KNOW,I)
      SE(I) = 0.0
  150 CONTINUE
      IF(ANISOT) CALL DCOEFF(KCO,NZ)
C     IF(SWBDIF.AND.ZBDIF.LT.1E-4) CALL OHM(KNOW,MLSWPZ)
      ZNCV = CV*RHO(KNOW,KNXZ)
      E1(KNXZ) = 1.0/ZNCV
      E2(KNXZ) = 0.0
      SE(KNXZ) = SHARE * (ES(KNOW,KNXZ) + SE(KNXZ))
      B1(KNXZ) = 1.0
      B2(KNXZ) = 0.0
      KM1 = KNXZ - 1
      DO 201 JZ=1,KM1
      ZNCV = CV*RHO(KNOW,JZ)
      E1(JZ) = 1.0/ZNCV
      E2(JZ) = (ZKAPPA(JZ) + ZKAPPA(JZ+1)) * 0.5
      SE( JZ ) = SHARE * (ES(KNOW, JZ ) + SE( JZ ))
      B1(JZ) = 1.0
      B2(JZ) = 0.5 * (ETA(JZ) + ETA(JZ+1))
  201 CONTINUE
      IF(LTEST) WRITE(10,9997)
 9997 FORMAT(16H          DIFUCO)
      RETURN
C
C         WHEN A ZERO OR NEGATIVE PRESSURE IS FOUND, THIS MAY GIVE
C         TROUBLES IN DCOEFF. USUALLY, THIS IS A MATTER OF INSTABILITY
C         OR BAD BOUNDARY CONDITIONS.
C         THE CELL WHERE P IS NONPOSITIVE, IS MENTIONED AND A STOP IS
C         FORCED LABELLED BY "STOP" 00001.
C
  300 CONTINUE
      KX = I
      KZ = KNOW
      IF(MLSWPZ) KX = KNOW
      IF(MLSWPZ) KZ = I
      WRITE(6,301) KX,KZ
  301 FORMAT(28HOZERO PRESSURE OCCURS AT KX= ,I3,10H AND KZ = ,I3)
      STOP 00001
      END
      SUBROUTINE DIFUSE(KNOW,KSTART,KEND,PD,KTORB)
      COMMON/DIFSRC/B1   (104),B2   (104),E1   (104),E2   (104),SE(104),
     2              AD   (104),BD   (104),CD   (104),DD   (104),
     3              TENEW(104),TE   (104),BNEW (104),B    (104),
     4              ETA  (104),KAPPAB(104),ZKAPPA(104),MLSWPZ
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
```

```
     2              BX   (40,70),BZ   (40,70),U    (40,70),
     3              ES   (40,70),A    (40,70), TIME, MEMDUM,
     4              REYN,BETA,VINP
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/METRIC/ X(104),Z(104),ORHO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/BCAD/SWIMPL
      DIMENSION XZ(104)
      REAL KAPPAB,MEMDUM
      LOGICAL MLSWPZ,LTEST,LPRINT,LINIT,SWIMPL
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         THE COEFFICIENTS OF THE TRIDIAGONAL MATRIX EQUATION
C         D P / D T = AD(J)*P(J-1)+BD(J)*P(J)+CD(J)*P(J+1)+DD(J)
C         ARE DETERMINED AND SOLVED FOR P(J).
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
      DO 120 L = KSTART,KEND
      IF(MLSWPZ)XZ(L) = Z(L)
  120 IF(.NOT.MLSWPZ)XZ(L) = X(L)
C         1.   DIFFUSION OF TEMPERATURE
      IF(KTORB.EQ.2) GOTO 200
  100 CONTINUE
      ZE2N = 0.0
      DO 101 J=KSTART,KEND
      ZE2P = E2(J)
      IF(J.EQ.KEND) ZE2P = 0.0
      IF(J.EQ.KEND) GO TO 300
      IF(J.EQ.KSTART) GO TO 310
      AZALPH = 2.0*E1(J)/((XZ(J+1)-XZ(J-1))*(XZ(J)-XZ(J-1)))
      BZALPH = 2.0*E1(J)/((XZ(J)-XZ(J-1))*(XZ(J+1)-XZ(J)))
      CZALPH = 2.0*E1(J)/((XZ(J+1)-XZ(J-1))*(XZ(J+1)-XZ(J)))
      GO TO 330
  300 AZALPH = 1.0*E1(KEND)/((XZ(KEND)-XZ(KEND-1))**2)
      BZALPH = 2.0*E1(KEND)/((XZ(KEND)-XZ(KEND-1))**2)
      CZALPH = 1.0*E1(KEND)/((XZ(KEND)-XZ(KEND-1))**2)
      GO TO 330
  310 AZALPH = 2.0*E1(1)/(XZ(4)*(XZ(4)+XZ(3)))
      BZALPH = 2.0*E1(1)/(XZ(3)*(XZ(4)-XZ(3)))
      CZALPH = 2.0*E1(1)/(XZ(4)*XZ(3))
  330 CONTINUE
      AD(J) = AZALPH * ZE2N
      BD(J) = -BZALPH * (ZE2N+ZE2P)
      CD(J) = CZALPH * ZE2P
      ZE2N = ZE2P
      DD(J) = SE(J)
  101 CONTINUE
C         IMPLICIT BOUNDARY CONDITIONS
      IF(.NOT.SWIMPL) GO TO 110
      KENDM1 = KEND - 1
      CD(2) = AD(2) + CD(2)
      AD(KENDM1) = AD(KENDM1) + CD(KENDM1)
  110 CONTINUE
C
C         THE TRIDIAGONAL MATRIX IS SOLVED TO FIND THE DIFFUSED VALUE.
C
```

```
      CALL SOLVE(KNOW,KEND,1)
      IF(LTEST) WRITE(10,9999)
      RETURN
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         2.  DIFFUSION OF MAGNETIC FIELD
C
  200 CONTINUE
      ZB2N = 0.0
      DO 201 J=KSTART,KEND
      ZB2P = B2(J)
      IF(J.EQ.KEND) ZB2P = 0.0
      IF(J.EQ.KEND) GO TO 400
      IF(J.EQ.KSTART) GO TO 410
      AZALPH = 2.0*B1(J)/((XZ(J+1)-XZ(J-1))*(XZ(J)-XZ(J-1)))
      BZALPH = 1.0*B1(J)/((XZ(J)-XZ(J-1))*(XZ(J+1)-XZ(J)))
      CZALPH = 2.0*B1(J)/((XZ(J+1)-XZ(J-1))*(XZ(J+1)-XZ(J)))
      GO TO 430
  400 AZALPH = 1.0*B1(KEND)/((XZ(KEND)-XZ(KEND-1))**2)
      BZALPH = 1.0*B1(KEND)/((XZ(KEND)-XZ(KEND-1))**2)
      CZALPH = 1.0*B1(KEND)/((XZ(KEND)-XZ(KEND-1))**2)
      GO TO 430
  410 AZALPH = 2.0*B1(1)/(XZ(4)*(XZ(4)-XZ(3)))
      BZALPH = 1.0*B1(1)/(XZ(3)*(XZ(4)-XZ(3)))
      CZALPH = 2.0*B1(1)/(XZ(4)*XZ(3))
  430 CONTINUE
      AD(J) = AZALPH * ZB2N
      BD(J) = -BZALPH * (ZB2N+ZB2P)
      CD(J) = CZALPH * ZB2P
      ZB2N = ZB2P
      DD(J) = 0.0
  201 CONTINUE
C         IMPLICIT BOUNDARY CONDITIONS
      IF(.NOT.SWIMPL) GO TO 210
      KENDM1 = KEND - 1
      IF(.NOT.MLSWPZ) CD(2) = CD(2) + AD(2)
      GRX = (XZ(KEND)-XZ(KENDM1))/(XZ(KENDM1)-XZ(KEND-2))
      IF(.NOT.MLSWPZ) AD(KENDM1) = AD(KENDM1) - GRX*CD(KENDM1)
      IF(.NOT.MLSWPZ) BD(KENDM1) = BD(KENDM1) + (1.+GRX)*CD(KENDM1)
      GRZ = (XZ(1)-XZ(2))/(XZ(2)-XZ(3))
      IF(MLSWPZ) CD(2) = CD(2) + AD(2)
C         NEXT STATEMENT APPLYS `F. F.' CONDITION TO BOTTOM SURFACE
C     IF(MLSWPZ) AD(KENDM1) = AD(KENDM1) + CD(KENDM1)
  210 CONTINUE
      CALL SOLVE(KNOW,KEND,2)
      IF(LTEST) WRITE(10,9999)
 9999 FORMAT(16H          DIFUSE )
      RETURN
      END
      SUBROUTINE SOLVE (KNOW,KEND,KTORB)
      COMMON/DIFSRC/B1   (104),B2   (104),E1   (104),E2   (104),SE(104),
     2              AD   (104),BD   (104),CD   (104),DD   (104),
     3              TENEW(104),TE   (104),BNEW (104),B    (104),
     4              ETA  (104),KAPPAB(104),ZKAPPA(104),MLSWPZ
      COMMON/FYSVAR/RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2              BX   (40,70),BZ   (40,70),U    (40,70),
```

```
     3              ES   (40,70),A    (40,70),TIME ,MEMDUM,
     4              REYN,BETA,VINP
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/BCAD/SWIMPL
      REAL KAPPAB,MEMDUM
      LOGICAL MLSWPZ,LTEST,LPRINT,LINIT,SWIMPL
      COMMON/TIMING/DT,DTMIN,DTMAX,AK0,AK1,AK2,AK3
      DIMENSION E(104),F(104)
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         SOLVE INVERTS THE TRIDIAGONAL MATRIX DESCRIBING THE DIFFUSION.
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         1. SOLVE THE DIFFUSIVE PART OF THE HEAT EQUATION.
      IA = 1
      IF(SWIMPL) IA = 2
      IF(KTORB.EQ.2) GOTO 200
  100 CONTINUE
      ZRDT = 0.0
      IF (DT.GT.0.0) ZRDT = 1.0/ DT
      ZE = ZRDT * TE(IA) + DD(IA)
      ZDEN = ZRDT - BD(IA)
      E(IA) = CD(IA) / ZDEN
      F(IA) = ZE / ZDEN
      KENDM1 = KEND - 1
      IAJ = IA + 1
      DO 110 L = IAJ,KENDM1
      ZE = ZRDT * TE(L) + DD(L)
      ZDEN = ZRDT - BD(L) - AD(L) * E(L-1)
      E(L) = CD(L) / ZDEN
      F(L) = (ZE + AD(L) * F(L-1)) / ZDEN
  110 CONTINUE
C         IF SWIMPL IS FALSE, THEN BOUNDARY CONDITION ASSIGNMENT
C         MAY CAUSE SIGNIFICANT ERROR FOR LARGE KAPPA.
      IF(SWIMPL) GO TO 130
      IX = 1
      ZE = ZRDT * TE(KEND) + DD(KEND)
      E(KEND) = E(KENDM1)
      F(KEND) = (ZE + AD(KEND) * F(KENDM1)) / ZDEN
C         1.2 FIND TENEW STARTING FROM OUTER BOUNDARY
      TEND = TE(KEND)
      TENEW (KEND) = E(KEND)*TEND + F(KEND)
      GO TO 140
  130 TENEW(1) = TE(1)
      TENEW(KEND) = TE(KEND)
      TENEW(KENDM1) = F(KENDM1)
      IX = 3
  140 CONTINUE
      DO 120 LDASH = IX,KENDM1
      L = KEND - LDASH + IA - 1
      TENEW (L) = E(L) * TENEW (L+1) + F(L)
  120 CONTINUE
      IF(LTEST) WRITE(10,9999)
      RETURN
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         2.  SOLVE THE MAGNETIC FIELD EQUATION
  200 CONTINUE
```

```
      ZRDT = 0.0
      IF (DT.GT.0.0) ZRDT = 1.0 / DT
      ZE = ZRDT * B(IA) + DD(IA)
      ZDEN = ZRDT - BD(IA)
      E(IA) = CD(IA) / ZDEN
      F(IA) = ZE / ZDEN
      KENDM1 = KEND - 1
      IAJ = IA + 1
      DO 210 L = IAJ,KENDM1
      ZE = ZRDT * B(L) + DD(L)
      ZDEN = ZRDT - BD(L) - AD(L) * E(L-1)
      E(L) = CD(L) / ZDEN
      F(L) = (ZE + AD(L) * F(L-1)) / ZDEN
  210 CONTINUE
C         IF SWIMPL FALSE, THEN BOUNDARY CONDITION ASSIGNMENT
C         MAY CAUSE SIGNIFICANT ERROR FOR LARGE ETA.
      IF(SWIMPL) GO TO 230
      IX = 1
      ZE = ZRDT * B(KEND) + DD(KEND)
      E(KEND) = E(KENDM1)
      F(KEND) = (ZE + AD(KEND) * F(KENDM1)) / ZDEN
C         2.2  FIND BNEW STARTING FROM OUTER BOUNDARY
      BEND = B(KEND)
      BNEW(KEND) = E(KEND)*BEND + F(KEND)
      GO TO 240
  230 BNEW(1) = B(1)
      BNEW(KEND) = B(KEND)
      BNEW(KENDM1) = F(KENDM1)
      IX = 3
  240 CONTINUE
C         NEXT STATEMENT EXTENDS SCAN TO INCLUDE KEND IN Z SCAN
      IF(SWIMPL.AND.MLSWPZ) IX = 2
      DO 220 LDASH = IX,KENDM1
      L = KEND - LDASH + IA - 1
      BNEW(L) = E(L) * BNEW(L+1) + F(L)
  220 CONTINUE
      IF(LTEST) WRITE(10,9999)
 9999 FORMAT(16H          SOLVE )
      RETURN
      E N D
      SUBROUTINE AOHM(KNOW,MLSWPZ)
      LOGICAL MLSWPZ
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/DIFSRC/A1 (104),A2 (104),A3 (104),A4 (104),SE (104),
     2              A6 (104),A7 (104),A8 (104),A9 (104),
     3              A10(104),A11(104),A12(104),A13(104),
     4              ETA(104)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C         THE DISSIPATION DUE TO OHMIC LOSSES IS CALCULATED IN ROW OR
C         COLUMN, IF THE RESISTIVITY ETA IS AT LEAST FOUR ORDERS OF
C         MAGNITUDE LARGER THAN ITS CLASSICAL VALUE (ELSE THE EFFECT
```

```
C         IS NEGLIGIBLE). THE DISSIPATION DISS = ETA*(CURL(B))**2 IS
C         INCLUDED IN THE DIFFUSIONAL SOURCE OF THE HEAT DIFFUSION
C         EQUATION.
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C         SCAN IN X-DIRECTION: ROW.
C
      TDX = 2.*DX
      TDZ = 2.*DZ
      NXM1 = NX-1
      NZM1 = NZ-1
      IF(MLSWPZ) GOTO 2
      J = KNOW
      IF(J.EQ.1) J=2
      IF(J.EQ.NZ) J=NZ-1
      DO 1 I=2,NXM1
      CURR = (BX(I,J+1)-BX(I,J-1))/TDZ - (BZ(I+1,J)-BZ(I-1,J))/TDX
      DISS = CURR*CURR*ETA(I)
      SE(I) = DISS
    1 CONTINUE
      SE(1) = SE(2)
      SE(NX) = SE(NX-1)
      RETURN
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C         SCAN IN Z-DIRECTION: COLUMN
C
    2 CONTINUE
      I = KNOW
      IF(I.EQ.1) I=2
      IF(I.EQ.NX) I=NX-1
      DO 3 J=2,NZM1
      CURR = (BX(I,J+1)-BX(I,J-1))/TDZ - (BZ(I+1,J)-BZ(I-1,J))/TDX
      DISS = CURR*CURR*ETA(J)
      SE(J) = DISS
    3 CONTINUE
      SE(1) = SE(2)
      SE(NZ) = SE(NZ-2)
      RETURN
      END
      SUBROUTINE SSETYY(PFOLD,PFNEW,KZ,KSTART,KEND)
      DIMENSION PFOLD(KZ),PFNEW(KZ),ZRDHO(104)
      COMMON/SHARYY/ZDHOLD(104),ZNH(104),ZDHNEW(104),ZDNEW(104),ZRDNEW
     R (104),ZDPQOD(104),ZDMQOD(104),ZRDHC(104),ZDTO(104),ZDTDHC(104),
     R ZCH(104),ZDPH(104),ZDPOCH(104),ZDMOCH(104),ZDMH(104)
     R ,ZDTH
C         1. SET ALL VELOCITY INDEPENDENT CONSTANTS
      J1 = KSTART + 1
      J1P1 = J1 + 1
      J2 = KEND - 1
      J2M1 = J2 - 1
      DO 101 J=KSTART,J2
      ZDHOLD(J) = PFOLD(J+1) - PFOLD(J)
      ZRDHO(J) = 1.0 / ZDHOLD(J)
      ZNH(J) = 0.5 * (PFNEW(J+1) + PFNEW(J))
```

```
      ZDHNEW(J) = PFNEW(J+1) - PFNEW(J)
  101 CONTINUE
      DO 102 J=J1P1,J2M1
      ZDNEW(J) = ZDHNEW(J) + ZDHNEW(J-1)
      ZRDNEW(J) = 1.0 / ZDNEW(J)
  102 CONTINUE
      ZDNEW(J1) = ZDHNEW(J1) + ZDHNEW(KSTART)
      ZRDNEW(J1) = 1.0 / ZDNEW(J1)
      ZDNEW(J2) = ZDHNEW(J2M1) + ZDHNEW(J2)
      ZRDNEW(J2) = 1.0 / ZDNEW(J2)
      DO 103 J=J1,J2
      ZDPQOD(J) = 0.25 * ZDHOLD(J) * ZRDNEW(J)
      ZDMQOD(J) = 0.25 * ZDHOLD(J-1) * ZRDNEW(J)
  103 CONTINUE
C     WRITE(6,9997)
 9997 FORMAT(16H          SSETYY)
      RETURN
      E N D
      SUBROUTINE SSETVY(PFOLD,PVC,KZ,KSTART,KEND,PDT)
      DIMENSION PFOLD(KZ),PVC(KZ),ZC(104),ZDHC(104)
      COMMON/SHARYY/ZDHOLD(104),ZNH(104),ZDHNEW(104),ZDNEW(104),ZRDNEW
     R (104),ZDPQOD(104),ZDMQOD(104),ZRDHC(104),ZDTO(104),ZDTDHC(104),
     R ZCH(104),ZDPH(104),ZDPOCH(104),ZDMOCH(104),ZDMH(104)
     R ,ZDTH
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         2.  SET VELOCITY DEPENDENT COEFFICIENTS HERE
      J1= KSTART+1

      J2= KEND-1
      ZDTH = 0.5 * PDT
      DO 201 J=KSTART,KEND
  700 ZC(J) = PFOLD(J) + PVC(J)*PDT
  201 CONTINUE
      DO 202 J=KSTART,J2
      ZDHC(J) = ZC(J+1) - ZC(J)
  701 ZRDHC(J) = 1.0 / ZDHC(J)
      ZDTO(J) = PDT * ZRDHC(J)
  703 ZDTDHC(J) = ZDTH * ZRDHC(J)
      ZCH(J) = ZDHOLD(J) * ZRDHC(J)
  704 ZDPH(J) = ZC(J+1) - ZNH(J)
      ZDPOCH(J) = ZDPH(J) * ZRDHC(J)
  705 ZDMH(J) = ZNH(J) - ZC(J)
      ZDMOCH(J) = ZDMH(J) * ZRDHC(J)
  202 CONTINUE
      RETURN
      E N D
      SUBROUTINE SHASTY(PFOLD,PFNEW,KLG,PG,KLS,PS,PH,KLQ,PQ,KZ,KSTART,
     R KEND,PRBC)
      DIMENSION PFOLD(KZ),PFNEW(KZ),PG(KZ),PS(KZ),PH(KZ),PQ(KZ),ZDELH
     R (104),ZFLUXH(104),ZFRD(104),ZRHS(104),ZFPH(104),ZFMH(104),
     R ZFCH(104),ZDFPH(104),ZDFMH(104),ZFTD(104)
      COMMON/BOUNDI/   RULX,RULZ
      COMMON/SHARYY/ZDHOLD(104),ZNH(104),ZDHNEW(104),ZDNEW(104),ZRDNEW
     R (104),ZDPQOD(104),ZDMQOD(104),ZRDHC(104),ZDTO(104),ZDTDHC(104),
     R ZCH(104),ZDPH(104),ZDPOCH(104),ZDMOCH(104),ZDMH(104)
```

```
     R ,ZDTH
      LOGICAL KLG, KLQ, KLS
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C          THE DIFFUSION STAGE
      ZPBC = 1.0 - PRBC**2
      J1= KSTART+1
C     IF (PRBC.NE.1.0) PQ(1) = 0.0
C     IF (PRBC.NE.1.0) PQ(2) = 0.0
      J2= KEND-1
      J1P1=J1+1
      J2M1=J2-1
C          3.1  COMPUTE DIFFUSED VALUES OF F
      ZDELH(KSTART) = PFOLD(J1) - PFOLD(KSTART)
      ZFLUXH(KSTART) = 0.25 * ZDHNEW(KSTART) * ZDELH(KSTART)
      DO 311 J=J1,J2
      ZDELH(J) = PFOLD(J+1) - PFOLD(J)
      ZFLUXH(J) = 0.25 * ZDHNEW(J) * ZDELH(J)
      ZFRD(J) = PFOLD(J) + ZDPQOD(J)*ZDELH(J) - ZDMQOD(J)*ZDELH(J-1)
  311 CONTINUE
C          3.2  APPLY BOUNDARY CONDITIONS
      ZDELH(KEND) = ZPBC*ZDELH(J1) - ABS(PRBC)*ZDELH(KEND-3)
      ZFLUXH(KEND) = -ABS(PRBC)*ZFLUXH(KEND-3)
C     ZFRD(KSTART) =       ZFRD(J1)
      ZFRD(KSTART) = PRBC*ZFRD(J1P1)
      ZFRD(KEND  ) = ABS(PRBC)* ZFRD(J2M1)
C     ZFRD(KEND  ) = ABS(PRBC) * ZFRD(J2)
      ZFLUXH(J2  ) = -ABS(PRBC)*ZFLUXH(J2M1)
      DO 331 J=KSTART,J2
  331 ZRHS(J) = 0.0
C          3.3  CALCULATE NON-ZERO SOURCE TERMS
      IF(KLG.AND.KLQ.AND.KLS) GO TO 340
      IF(KLG) GO TO 333
      DO 332 J=KSTART,J2
  332 ZRHS(J) = ZRHS(J) + ZDTO(J) * (PG(J+1) - PG(J))
  333 IF (KLS) GO TO 335
      DO 334 J=KSTART,J2
  334 ZRHS(J) = ZRHS(J) + ZDTDHC(J) * (PS(J+1) + PS(J))
     +                               * (PH(J+1) - PH(J))
  335 IF(KLQ) GO TO 340
      DO 336 J=KSTART,J2
  336 ZRHS(J) = ZRHS(J) + (PQ(J)+PQ(J+1))*ZDHOLD(J)*ZRDHC(J)*ZDTH
C          3.4  CONVECT OLD FUNCTION VALUES
  340 CONTINUE
      ZRHS(KEND)=ZRHS(J2M1)
      DO 341 J=KSTART,J2
      ZFPH(J) = ZCH(J) * PFOLD(J+1)
  341  ZFMH(J) = ZCH(J) * PFOLD(J)
C          3.5  ADD SOURCE TERMS TO LEFT & RIGHT HALF-CELLS
      IF(KLG.AND.KLQ.AND.KLS) GO TO 352
      DO 351 J=KSTART,J2

      ZFPH(J) = ZFPH(J) + ZRHS(J)
  351 ZFMH(J) = ZFMH(J) + ZRHS(J)
  352 CONTINUE
      DO 353 J=KSTART,J2
```

```
      ZFCH(J) = ZFPH(J) * ZDMOCH(J) + ZFMH(J) * ZDPOCH(J)
      ZDFPH(J) = ZDPH(J) * (ZFPH(J) + ZFCH(J))
  353 ZDFMH(J) = ZDMH(J) * (ZFCH(J) + ZFMH(J))
      ZFMH(KEND) = PRBC*ZDHNEW(J2)*ZRDHC(KEND)*PFOLD(J2M1)+ZRHS(KEND)
      ZFPH(KEND) = PRBC*ZDHNEW(J2)*PFOLD(KEND-3)+ZRHS(KEND)
      ZFCH(KEND) = ZDMH(KEND)*ZRDHC(KEND)*ZFPH(KEND)
     + + ZDPH(KEND)*ZRDHC(KEND)*ZFMH(KEND)
C         3.6  REAPPORTATION NEW F'S TO NEW CELLS
      DO 361 J=J1,J2
      ZFTD(J) = ZRDNEW(J) * (ZDFMH(J) + ZDFPH(J-1))
  361 CONTINUE
C     ZFTD(KSTART) =       ZFTD(J1)
      ZFTD(KSTART) = PRBC*ZFTD(J1P1)
C     ZFTD (KEND)=(ZDFMH(KEND)+ZDFPH(KEND))*ZRDNEW(J2)
      ZFTD(KEND  ) = ABS(PRBC) * ZFTD(J2M1)
      ZDFPH(KEND)=ZDPH(KEND)*(ZFPH(KEND)+ZFCH(KEND))
      ZDFMH(KEND)=ZDMH(KEND)*(ZFCH(KEND)+ZFMH(KEND))
C - -  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         4.  FLUX LIMITING & ANTIDIFFUSION
C         4.1  CALCULATE FLUXES
      ZDFTP = ZFTD(KSTART) - ZFRD(KSTART)
      DO 411 J=KSTART,J2
      ZDFTI = ZDFTP
      ZDFTP = ZFTD(J+1) - ZFRD(J+1)
      ZDELH(J) = ZFTD(J+1) - ZFTD(J)
  411 ZFLUXH(J) = ZFLUXH(J) + 0.25 * ZDHNEW(J) * (ZDFTP - ZDFTI)
C         4.2  LIMIT FLUXES TO AVOID EXTREMA
      DO 421 J=J1,J2M1
      ZSGN = SIGN(1.0 , ZFLUXH(J))
  421 ZFLUXH(J) = ZSGN * AMAX1( 0.0 , AMIN1(ZSGN*ZDNEW(J)*ZDELH(J-1),
     +                                ABS(ZFLUXH(J)),
     +                                ZSGN*ZDNEW(J+1)*ZDELH(J+1)))
      ZSGN = SIGN(1.0 , ZFLUXH(J2))
      ZDELHO = ZPBC*ZDELH(J1)*ZDNEW(J1) - PRBC*ZDELH(J2M1)*ZDNEW(J2)
C     ZFLUXH(J2) = ZSGN * AMAX1(0.0 , AMIN1(ZSGN*ZDNEW(J2)*
C    +             ZDELH(J2M1),ABS(ZFLUXH(J2)),ZSGN*ZDELHO))
      ZSGN = SIGN(1.0 , ZFLUXH(KSTART))
C     ZDELHO =-ZDELH(J1)*ZDNEW(J1)
      ZDELHO = -PRBC*ZDELH(J1P1)*ZDNEW(J1P1)
C     ZFLUXH(KSTART) = ZSGN * AMAX1(0.0 , AMIN1(ZSGN*ZDELHO,
C    +                 ABS(ZFLUXH(KSTART)),ZSGN*ZDNEW(J1)*ZDELH(J1)))
      ZFLUXH(KSTART) = -PRBC*ZFLUXH(J1)
      ZFLUXH(J2    ) = -ABS(PRBC)*ZFLUXH(J2M1)
C THIS IS THE FLUX FOR A FREE FLOATING BOUNDARY WHEN F(NX)=F(NX-1)
C         4.3  APPLY ANTI-DIFFUSION
      DO 431 J=J1,J2
  431 PFNEW(J) = ZFTD(J) - (ZFLUXH(J)-ZFLUXH(J-1)) * ZRDNEW(J)
      PFNEW(KEND) = ABS(PRBC)*PFNEW(J2M1)
C     IF(PRBC.LT.0.0) PFNEW(J2)=0.0
      PFNEW(KSTART) = PRBC*PFNEW(J1P1)
 2001 FORMAT(I5,8E11.3)
 2002 FORMAT(I5,4E11.3)
      RETURN
      END
      SUBROUTINE SSETBB(PFOLD,PFNEW,KZ,KSTART,KEND)
```

```
      DIMENSION PFOLD(KZ),PFNEW(KZ),ZRDHO(104)
      COMMON/SHARAB/ZDHOLD(104),ZNH(104),ZDHNEW(104),ZDNEW(104),ZRDNEW
     R (104),ZDPQOD(104),ZDMQOD(104),ZRDHC(104),ZDTO(104),ZDTDHC(104),
     R ZCH(104),ZDPH(104),ZDPOCH(104),ZDMOCH(104),ZDMH(104)
     R ,ZDTH
      J1 = KSTART + 1
      J1P1 = J1 + 1
      J2 = KEND - 1
      J2M1 = J2 - 1
      DO 101 J=KSTART,J2
      ZDHOLD(J) = PFOLD(J+1) - PFOLD(J)
      ZRDHO(J) = 1.0 / ZDHOLD(J)
      ZNH(J) = 0.5 * (PFNEW(J+1) + PFNEW(J))
      ZDHNEW(J) = PFNEW(J+1) - PFNEW(J)
  101 CONTINUE
      ZDHOLD(KEND) = ZDHOLD(J2)
      ZDHNEW(KEND) = ZDHNEW(J2)
      ZNH(KEND) = ZNH(J2) + ZDHOLD(J2)
      DO 102 J=J1P1,J2M1
      ZDNEW(J) = ZDHNEW(J) + ZDHNEW(J-1)
      ZRDNEW(J) = 1.0 / ZDNEW(J)
  102 CONTINUE
      ZDNEW(J1) = ZDHNEW(J1) + ZDHNEW(KSTART)
      ZRDNEW(J1) = 1.0 / ZDNEW(J1)
      ZDNEW(J2) = ZDHNEW(J2M1) + ZDHNEW(J2)
      ZRDNEW(J2) = 1.0 / ZDNEW(J2)
      DO 103 J=J1,J2
      ZDPQOD(J) = 0.25 * ZDHOLD(J) * ZRDNEW(J)
      ZDMQOD(J) = 0.25 * ZDHOLD(J-1) * ZRDNEW(J)
  103 CONTINUE
C     WRITE(6,9997)
 9997 FORMAT(16H           SSETBB)
      RETURN
      E N D
      SUBROUTINE SSETVB(PFOLD,PVC,KZ,KSTART,KEND,PDT)
      DIMENSION PFOLD(KZ),PVC(KZ),ZC(104),ZDHC(104)
      COMMON/SHARAB/ZDHOLD(104),ZNH(104),ZDHNEW(104),ZDNEW(104),ZRDNEW
     R (104),ZDPQOD(104),ZDMQOD(104),ZRDHC(104),ZDTO(104),ZDTDHC(104),
     R ZCH(104),ZDPH(104),ZDPOCH(104),ZDMOCH(104),ZDMH(104)
     R ,ZDTH
      J1= KSTART+1
      J2= KEND-1
      J2M1 = J2 - 1
      ZDTH = 0.5 * PDT
      DO 201 J=KSTART,KEND
      ZC(J) = PFOLD(J) + PVC(J)*PDT
  201 CONTINUE
      ZC (J2) = PFOLD(J2)
      ZC(KEND) = PFOLD(KEND) - PVC(J2M1)*PDT
      ZC(KEND+1) = PFOLD(KEND) + ZDHOLD(KEND)-PVC(KEND-3)*PDT
      DO 202 J=KSTART,J2
      ZDHC(J) = ZC(J+1) - ZC(J)
      IF(ZDHC(J).EQ.0.0) WRITE(6,100) J
  100 FORMAT(I5)
      ZRDHC(J) = 1.0 / ZDHC(J)
```

```
      ZDTO(J) = PDT * ZRDHC(J)
      ZDTDHC(J) = ZDTH * ZRDHC(J)
      ZCH(J) = ZDHOLD(J) * ZRDHC(J)
      ZDPH(J) = ZC(J+1) - ZNH(J)
      ZDPOCH(J) = ZDPH(J) * ZRDHC(J)
      ZDMH(J) = ZNH(J) - ZC(J)
      ZDMOCH(J) = ZDMH(J) * ZRDHC(J)
  202 CONTINUE
      ZDHC(KEND) = ZC(KEND+1) - ZC(KEND)
      ZRDHC(KEND) = 1.0/ZDHC(KEND)
      ZDMH(KEND) = ZNH(KEND) - ZC(KEND)
      ZDPH(KEND) = ZC(KEND+1) - ZNH(KEND)
      RETURN
      E N D
      SUBROUTINE SHASTB(PFOLD,PFNEW,KLG,PG,KLS,PS,PH,KLQ,PQ,KZ,KSTART,
     R KEND,PRBC,PRBCT)
      DIMENSION PFOLD(KZ),PFNEW(KZ),PG(KZ),PS(KZ),PH(KZ),PQ(KZ),ZDELH
     R (104),ZFLUXH(104),ZFRD(104),ZRHS(104),ZFPH(104),ZFMH(104),
     R ZFCH(104),ZDFPH(104),ZDFMH(104),ZFTD(104)
      COMMON/SHARAB/ZDHOLD(104),ZNH(104),ZDHNEW(104),ZDNEW(104),ZRDNEW
     R (104),ZDPQOD(104),ZDMQOD(104),ZRDHC(104),ZDTO(104),ZDTDHC(104),
     R ZCH(104),ZDPH(104),ZDPOCH(104),ZDMOCH(104),ZDMH(104)
     R ,ZDTH
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SWMHDB,SW2DIM
      LOGICAL KLG, KLQ, KLS
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         THE DIFFUSION STAGE
      ZPBC = 1.0-PRBC*PRBC
      IF(SWGRAV.AND.PRBCT.NE.1.0) PQ(1) = 0.0
      IF(SWGRAV.AND.PRBCT.NE.1.0) PQ(2) = 0.0
      ZPBCT = 1.0 - PRBCT*PRBCT
      J1= KSTART+1
      J2= KEND-1
      J1P1=J1+1
      J2M1=J2-1
C         3.1 COMPUTE DIFFUSED VALUES OF F
      ZDELH(KSTART) = PFOLD(J1) - PFOLD(KSTART)
      ZFLUXH(KSTART) = 0.25 * ZDHNEW(KSTART) * ZDELH(KSTART)
      DO 311 J=J1,J2
      ZDELH(J) = PFOLD(J+1) - PFOLD(J)
      ZFLUXH(J) = 0.25 * ZDHNEW(J) * ZDELH(J)
      ZFRD(J) = PFOLD(J) + ZDPQOD(J)*ZDELH(J) - ZDMQOD(J)*ZDELH(J-1)
  311 CONTINUE
C         3.2 APPLY BOUNDARY CONDITIONS
      ZFRD(KSTART) = ZPBC*ZFRD(J2) + PRBC*ZFRD(J1P1)
C         OLD B.C.'S
C     ZDELH(KEND) =ZPBCT*ZDELH(J1) -PRBCT*ZDELH(J2M1)
C     ZFLUXH(J2) = -PRBCT*ZFLUXH(J2M1)
C     ZFLUXH(KEND)= -PRBCT*ZFLUXH(KEND-3)
C     ZFRD(KEND) = PRBCT*ZFRD(J2M1)
C         B.C.'S FOR FIXED VALUE
      ZFLUXH(KEND) = 0.0
      ZFRD(KEND) = PFOLD(KEND)
      DO 331 J=KSTART,J2
```

```
  331 ZRHS(J) = 0.0
C         3.3 CALCULATE NON-ZERO SOURCE TERMS
      IF(KLG.AND.KLQ.AND.KLS) GO TO 340
      IF(KLG) GO TO 333
      DO 332 J=KSTART,J2
  332 ZRHS(J) = ZRHS(J) + ZDTO(J) * (PG(J+1) - PG(J))
  333 IF (KLS) GO TO 335
      DO 334 J=KSTART,J2
  334 ZRHS(J) = ZRHS(J) + ZDTDHC(J) * (PS(J+1) + PS(J))
     +                               * (PH(J+1) - PH(J))
  335 IF(KLQ) GO TO 340
      DO 336 J=KSTART,J2
  336 ZRHS(J) = ZRHS(J) + (PQ(J)+PQ(J+1))*ZDHOLD(J)*ZRDHC(J)*ZDTH
C         3.4 CONVECT OLD FUNCTION VALUES
  340 CONTINUE
      ZRHS(KEND) = ZRHS(J2M1)
      DO 341 J=KSTART,J2
      ZFPH(J) = ZCH(J) * PFOLD(J+1)
  341  ZFMH(J) = ZCH(J) * PFOLD(J)
C         3.5 ADD SOURCE TERMS TO LEFT & RIGHT CELLS
      IF(KLG.AND.KLQ.AND.KLS) GO TO 352
      DO 351 J=KSTART,J2
      ZFPH(J) = ZFPH(J) + ZRHS(J)
  351 ZFMH(J) = ZFMH(J) + ZRHS(J)
  352 CONTINUE
      DO 353 J=KSTART,J2
      ZFCH(J) = ZFPH(J) * ZDMOCH(J) + ZFMH(J) * ZDPOCH(J)
      ZDFPH(J) = ZDPH(J) * (ZFPH(J) + ZFCH(J))
  353 ZDFMH(J) = ZDMH(J) * (ZFCH(J) + ZFMH(J))
      ZFMH(KEND) = PRBCT*ZDHNEW(J2)*ZRDHC(KEND)*PFOLD(J2M1)+ZRHS(KEND)
      ZFPH(KEND) = PRBCT*ZDHNEW(J2)*PFOLD(KEND-3)+ZRHS(KEND)
      ZFCH(KEND) = ZDMH(KEND)*ZRDHC(KEND)*ZFPH(KEND)
     * + ZDPH(KEND)*ZRDHC(KEND)*ZFMH(KEND)
C         3.6 REAPPORTION NEW F'S TO NEW CELLS
      DO 361 J=J1,J2
  361 ZFTD(J) = ZRDNEW(J) * (ZDFMH(J) + ZDFPH(J-1))
      ZFTD(KSTART) = ZPBC*ZFTD(J2) + PRBC*ZFTD(J1P1)
C         OLD B.C.
C     ZDFMH(KEND) = ZDMH(KEND)*(ZFCH(KEND)+ZFMH(KEND))
C     ZDFPH(KEND) = ZDPH(KEND)*(ZFPH(KEND) + ZFCH(KEND))
C     ZFTD(KEND) = (ZDFMH(KEND) + ZDFPH(KEND))*ZRDNEW(J2)
C         B.C. FOR FIXED VALUE (PRIMARY ALTERATION SITE - PROPAGATES
C         BOUNDARY VALUE INTO TO INTERNAL MESH POINTS.)
      ZFTD(KEND) = PFOLD(KEND)
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         4. FLUX LIMITING & ANTI-DIFFUSION
C         4.1 CALCULATE FLUXES
      ZDFTP = ZFTD(KSTART) - ZFRD(KSTART)
      DO 411 J=KSTART,J2
      ZDFTI = ZDFTP
      ZDFTP = ZFTD(J+1) - ZFRD(J+1)
      ZDELH(J) = ZFTD(J+1) - ZFTD(J)
  411 ZFLUXH(J) = ZFLUXH(J) + 0.25 * ZDHNEW(J) * (ZDFTP - ZDFTI)
C         4.2 LIMIT FLUXES TO AVOID EXTREMA
      DO 421 J=J1,J2M1
```

```
      ZSGN = SIGN(1.0 , ZFLUXH(J))
  421 ZFLUXH(J) = ZSGN * AMAX1( 0.0 , AMIN1(ZSGN*ZDNEW(J)*ZDELH(J-1),
     +                                ABS(ZFLUXH(J)),
     +                                ZSGN*ZDNEW(J+1)*ZDELH(J+1)))
      ZSGN = SIGN(1.0 , ZFLUXH(J2))
C          OLD B.C.
C     ZDELHO =ZPBCT*ZDELH(J1)*ZDNEW(J1) -PRBCT*ZDELH(J2M1)*ZDNEW(J2)
C         B.C. FOR FIXED VALUE (SECONDARY ALTERATION SITE - REQUIRED
C         ONLY FOR FLUX LIMITING.)
      ZDELHO = 0.0
      ZFLUXH(J2) = ZSGN * AMAX1(0.0 , AMIN1(ZSGN*ZDNEW(J2)*
     +             ZDELH(J2M1),ABS(ZFLUXH(J2)),ZSGN*ZDELHO))
      ZSGN = SIGN(1.0 , ZFLUXH(KSTART))
      ZDELHO = ZPBC*ZDELH(J2M1)*ZDNEW(J2) - PRBC*ZDELH(J1)*ZDNEW(J1)
      ZFLUXH(KSTART) = ZSGN * AMAX1(0.0 , AMIN1(ZSGN*ZDELHO,
     +                 ABS(ZFLUXH(KSTART)),ZSGN*ZDNEW(J1)*ZDELH(J1)))
C         4.3 APPLY ANTI-DIFFUSION
      DO 431 J=J1,J2
  431 PFNEW(J) = ZFTD(J) - (ZFLUXH(J)-ZFLUXH(J-1)) * ZRDNEW(J)
C         OLD B.C.
      PFNEW(KEND) = PRBCT*PFNEW(J2M1)
C     IF(PRBCT.LT.0.0) PFNEW(J2) = 0.0
C         B.C. FOR FIXED VALUE (REDUNDANT IF SUBROUTINE BOUNDY IS
C         NEXT CALL.)
      PFNEW(KEND) = PFOLD(KEND)
      PFNEW(KSTART) = ZPBC*PFNEW(KEND-1) + PRBC*PFNEW(J1P1)
      RETURN
      END
      SUBROUTINE ADVECO(PA,PAVX,PAVZ,PE,PBX,PBZ,K,LSWEPZ,KNOW)
      DIMENSION PA(K),PAVX(K),PAVZ(K),PBX(K),PBZ(K),PE(K)
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SWMHDB,SW2DIM
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
      COMMON/SYSCON/ RSC,ANISOT,ZBDIF,ZKDIF,CPLRAD,PSI
      COMMON/MODO  /MODUS,SWTERM(6,3)
      COMMON/TESTER/LTEST,LPRINT,LINIT
      LOGICAL SWTERM,LSWEPZ,LTEST,LPRINT,LINIT,ANISOT
C         THIS ROUTINE SETS THE COEFFICIENTS IN THE TRANSPORT EQUATION
C         WHICH IS WRITTEN IN THE STANDARD FORM
C         DF/DT = DIV G + S.GRAD H + Q. ADVECO DETERMINES V,G,S,H,Q
C         SEE DESCRIPTION OF PROGRAM FOR TERMS DEFINED HERE FOR THE
C         EXAMPLES OF MODUS 1 AND 2.
C
      XREF = 1.0/RSC
      GAMM1 = 5./3.-1.
      IF(LSWEPZ) GOTO 200
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         SCAN IN X-DIRECTION. GRAVITATION AND SHEET AXIS ARE IN
```

```
C          THIS DIRECTION.
C
      DO 101 J=1,K
      JR = J
      JZ = KNOW
      IF(PE(J).EQ.0.0)PE(J) = 1.0E-15
      GKONST = 5.317*PA(J)/(PE(J)*RSC)
      IF (PA(J).EQ.0.0)PA(J) = 1.0E-4
      VELOCH(J) = PAVX(J)/PA(J)
      SSVR(J) = -1.0
      SHVR(J) = PE(J)
      SSVZ(J) = 0.0
      SHVZ(J)=0.0
      IF(.NOT.SWMHDB) GOTO 108
      SHVR(J) = SHVR(J) + 0.5*PBZ(J)**2
      IF(.NOT.SW2DIM) GOTO 108
      SSVZ(J) = PBX(J)
      SHVZ(J) = PBZ(J)
  108 CONTINUE
      SQVR(J) = 0.0
      IF(SWGRAV) SQVR(J) = -PA(J)*GKONST/(X(J)+XREF)**2
      IF(J.EQ.1) SQVR(J) = -SQVR(1)
      IF(.NOT.SWTEMP.AND..NOT.SWBDIF) GOTO 115
      SSEQ(J) = -GAMM1*PE(J)
      SQEQ(J) = 0.0
      IF(SWBDIF.AND.SWMHDB)SQEQ(J) = 0.5*GAMM1*OHM(JR,JZ)
      SHEQ(J) = VELOCH(J)
  115 CONTINUE
      IF(.NOT.SWMHDB.OR..NOT.SW2DIM) GOTO 126
      SGBZ(J) = PAVZ(J)*PBX(J)/PA(J)
  126 CONTINUE
  101 CONTINUE
C
C          IMPOSE BOUNDARY CONDITIONS ON VELOCITY OF CONVECTION
C
C     CALL BOUNDY(6,1)
      IF(LTEST) WRITE(10,9999)
      RETURN
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C          SCAN IN Z-DIRECTION.
C          SKIPPED IN ONE-DIMENSIONAL APPLICATIONS.
C
  200 CONTINUE
123   FORMAT(I4,2E10.3)
      DO 202 J = 1,K
C     PRINT 123,J,SHVZ(J),PBX(J)
202   CONTINUE
      DO 201 J = 1,K
      JR = KNOW
      IF (PA(J).EQ.0.0)PA(J)=1.0E-4
      JZ = J
      VELOCH(J) = PAVZ(J)/PA(J)
      SSVR(J) = 0.0
      SHVR(J) = 0.0
      SQVR(J) = 0.0
```

```
      IF(.NOT.SWMHDB) GOTO 205
      SSVR(J) = PBZ(J)
      SHVR(J) = PBX(J)
  205 CONTINUE
      SSVZ(J) = -1.0
      SHVZ(J) = PE(J)
      IF(.NOT.SWMHDB) GOTO 209
      SHVZ(J) = SHVZ(J) + 0.5*PBX(J)**2
  209 CONTINUE
      IF(.NOT.SWTEMP.AND..NOT.SWBDIF) GOTO 215
      SSEQ(J) = -GAMM1*PE(J)
      SQEQ(J) = 0.0
      IF(SWBDIF.AND.SWMHDB)SQEQ(J) = 0.5*GAMM1*OHM(JR,JZ)
      SHEQ(J) = VELOCH(J)
  215 CONTINUE
      IF(.NOT.SWMHDB) GOTO 226
      IF(PA(J).EQ.0.0)PA(J)=1.E-4
      SGBR(J) = PAVX(J)*PBZ(J)/PA(J)
  226 CONTINUE
  201 CONTINUE
C
C         IMPOSE BOUNDARY CONDITIONS ON VELOCITY OF CONVECTION
C
C     CALL BOUNDY(6,2)
      IF(LTEST) WRITE(10,9999)
 8002 FORMAT(2I3,7E15.4)
 9999 FORMAT(16H          ADVECO )
      RETURN
      END
      FUNCTION OHM(I,J)
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/SYSCON/RSC,ANISOT,ZBDIF,ZKDIF,CPLRAD,PSI
      ETA = 1./ZBDIF
      IF((I.EQ.1).OR.(I.EQ.NX)) GO TO 10
      CJ1 = (BZ(I+1,J)-BZ(I-1,J))/(X(I+1)-X(I-1))
      GO TO 30
   10 IF(I.EQ.NX) GO TO 20
      CJ1 = (BZ(2,J)+BZ(4,J))/(X(2)+X(4))
      GO TO 30
   20 CJ1 = (BZ(NX-3,J)-BZ(NX-1,J))/(X(NX)-X(NX-2))
   30 CONTINUE
      IF((J.EQ.1).OR.(J.EQ.NZ)) GO TO 40
      CJ2 = (BX(I,J+1)-BX(I,J-1))/(Z(J+1)-Z(J-1))
      GO TO 50
   40 IF(J.EQ.NZ) GO TO 60
      CJ2 = (BX(I,2)-BX(I,4))/(Z(3)-Z(1))
      GO TO 50
   60 CJ2 = (BX(I,NZ)-BX(I,NZ-1))/(Z(NZ)-Z(NZ-1))
   50 CONTINUE
      CJ = CJ1 - CJ2
      OHM = ETA*CJ*CJ
```

```
C     PRINT 70, I,J,CJ,OHM,ETA
C  70 FORMAT(1X,'I,J,CJ,OHM,ETA',2I3,3E15.6)
      RETURN
      END
      SUBROUTINE VTMSTP
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/TIMING/DT,DTMIN,DTMAX,AKO,AK1,AK2,AK3
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB
      REAL MEMDUM
      DT = 1.0E20
      DTL = DT
      DO 10 J = 1,NZ-1
      DO 10 I = 2,NX-1
      DXL = ABS(X(I+1) - X(I))
      IF(RHO(I,J).LE.0.0)RHO(I,J)=1.E-4
      ZRHO=1./RHO(I,J)
      IF(SWTEMP)CSQ = ABS((5./3.)*U(I,J)*ZRHO)
      IF(.NOT.SWTEMP)CSQ = ABS(U(I,J)*ZRHO)
      BAL = SQRT(BX(I,J)*BX(I,J) + BZ(I,J)*BZ(I,J))
      VAL = BAL*SQRT(ZRHO)
      VFML = SQRT(VAL*VAL + CSQ)
      VFL = SQRT((RHOVX(I,J)**2+RHOVZ(I,J)**2)*ZRHO)
      VL = VFML + VFL
      IF(VL.GT.0.0)DTL = AK1*DXL/VL
      DT = AMIN1(DTL,DT)
   10 CONTINUE
      RETURN
      END
```

Part Two: Boundary and initial conditions for the asymptotically steady-state problem

```
      SUBROUTINE BOUNDY(NO,JRZ)
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/MHDRZI/RHOI(104),RHOVXI(104),RHOVZI(104),BXI(104),BZI(104),
     2              UII   (104),
     3              RHOH(104),RHOVXH(104),RHOVZH(104),BXH(104),BZH(104),
     4              UHH   (104)
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB,
     1              SWOHMH
      COMMON/TIMING/DT,DTMIN,DTMAX,AK0,AK1,AK2,AK3
      COMMON/SYSCON/RSC,ANISOT,ZBDIF,ZKDIF,CPLRAD,PSI
      DIMENSION UO(104),RHOO(104),RHOVXO(104),AO(104),BXO(104),BZO(104)
     1,RHOVZO(104),DIVV(104),LINE(104)
      LOGICAL       SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB,
     1              SWOHMH
      REAL MEMDUM,SHEAR,DECR,PMIN,DELTA,GRX
      LOGICAL LTEST,LPRINT,LINIT,ANISOT
       GAMMA = 5./3.
       PI = 3.141
       if(time.le.2.0)vinp = 0.05*(time/2.0)
       if(time.ge.2.0)vinp = 0.05
C        B.C.'S FOR BASE (Z = 1.0)
      DO 10 I = 1,NX
      RHOVXO(I) = 0.0
      BXO(I) = 1.0/COS((REAL(I)*PI)/(REAL(NX)*10.0))
      if(time.le.2.)factor = time/2.
      if(time.ge.2.)factor = 1.0
      BZO(I) = (0.03+(factor*0.07))*SIN((REAL(I)*PI)/
     2(real(nx)*2.0))
      AO(I)=-1.0-DECR*VINP*TIME-((0.03+(factor*0.07))/(0.5*PI))*
     2COS((REAL(I)*PI)/(REAL(NX)*2.0))
      UO(I) = 0.05
      RHOO(I) = 1.0
      RHOVZO(I) = -DECR*VINP*RHOO(I)*
     2COS((REAL(I)*PI)/(REAL(NX)*10.0))
   10 CONTINUE

C         IMPOSE B.C.'S AT BASE CORNERS
      RHOO(1) = RHOO(3)
      RHOO(NX) = RHOO(NX-2)
      UO(1) = UO(3)
      UO(NX) = UO(NX-2)
      BZO(NZ) = BZO(NZ-2)
      GO TO (500,500,300,400,500,600),NO
```

```
  300 CONTINUE
C
C         ENTRY AFTER HALF TIME STEP IN X-CONVECTION
C
      IF(LTEST) WRITE(10,9993)
      BZH(NX) = BZH(NX-2)
      BZH(1) = -BZH(3)
      BXH(NX) = BX(NX,JRZ)
      BXH(1) = BXH(3)
      RHOH(NX) = RHOH(NX-2)
      RHOH(1) = RHOH(3)
      RHOVXH(NX) = RHOVXH(NX-2)
      RHOVXH(1) = -RHOVXH(3)
      RHOVZH(NX) = RHOVZH(NX-2)
      RHOVZH(1) = RHOVZH(3)
      UHH(NX) = UHH(NX-2)
      UHH(1) = UHH(3)
      RETURN
  400 CONTINUE
C
C         ENTRY AFTER HALF TIME STEP IN Z-CONVECTION
C
      IF(LTEST)WRITE(10,9994)
      BZH(NZ) = BZ0(JRZ)
      BZH(1) = BZH(3)
      BXH(NZ) = BX0(JRZ)
      RHOVXH(NZ) = RHOVX0(JRZ)
      RHOVZH(NZ) = RHOVZ0(JRZ)
      UHH(NZ) = U0(JRZ)
      RHOH(NZ) = RHO0(JRZ)
      BXH(1) = -BXH(3)
      RHOH(1) = RHOH(3)
      RHOVXH(1) = RHOVXH(3)
      RHOVZH(1) = -RHOVZH(3)
      UHH(1) = UHH(3)
      RETURN
  500 CONTINUE
C
C         ENTRY AFTER FULL TIME STEP IN DIFFUSION.
C         BOUNDARIES MUST BE MADE CONSISTENT FOR INPUT TO CONVECTION.
C
      IF(LTEST) WRITE(10,9995)
C         -------SURFACE 4-------BASE (INPUT FLOW)----------------------
      DO 128 I = 1,NX-1
      RHOVX(I,NZ) = RHOVX0(I)
      RHOVZ(I,NZ) = RHOVZ0(I)
      U(I,NZ) = U0(I)
      RHO(I,NZ) = RHO0(I)
      BX(I,NZ) = BX0(I)
      BZ(I,NZ) = BZ0(I)
      A(I,NZ) = A0(I)
  128 CONTINUE
C         -------SURFACE 1-------LEFT SIDE (AXIS OF SYMMETRY)----------
      DO 123 J = 2,NZ-1
      BX(1,J) = BX(3,J)
```

```
      BZ(1,J) = -BZ(3,J)
      BZ(2,J) = 0.0
      RHOVX(1,J) = -RHOVX(3,J)
      RHOVX(2,J) = 0.0
      RHOVZ(1,J) = RHOVZ(3,J)
      RHO(1,J) = RHO(3,J)
      A(1,J) = A(3,J)
  123 CONTINUE
C         -------SURFACE 2-------TOP (AXIS OF SYMMETRY)----------------
      DO 127 I = 1,NX-1
      BX(I,1) = -BX(I,3)
      BX(I,2) = 0.0
      BZ(I,1) = BZ(I,3)
      RHOVX(I,1) = RHOVX(I,3)
      RHOVZ(I,1) = -RHOVZ(I,3)
      RHOVZ(I,2) = 0.0
      RHO(I,1) = RHO(I,3)
      U(I,1) = U(I,3)
      A(I,1) = A(I,3)
  127 CONTINUE
C         -------SURFACE 3-------RIGHT SIDE (FREE-FLOATING)------------
      DO 124 J = 2,NZ-1
      BZ(NX,J) = BZ(NX-2,J)
      GR = (X(NX)-X(NX-2))/(Z(J+1)-Z(J-1))
      BX(NX,J) = BX(NX-2,J)-GR*(BZ(NX-1,J+1)-BZ(NX-1,J-1))
      RHOVX(NX,J) = RHOVX(NX-2,J)
      RHOVZ(NX,J) = RHOVZ(NX-2,J)
      RHO(NX,J) = RHO(NX-2,J)
      U(NX,J) = U(NX-2,J)
      A(NX,J) = A(NX-2,J) + BZ(NX,J)*(X(NX)-X(NX-2))
  124 CONTINUE
      BX(NX,1) = -BX(NX,3)
      BX(NX,2) = 0.0
      BZ(NX,1) = BZ(NX,3)
      RHOVX(NX,1) = RHOVX(NX-2,1)
      RHOVZ(NX,1) = RHOVZ(NX-2,1)
      RHO(NX,1) = RHO(NX-2,1)
      U(NX,1) = U(NX-2,1)
      GRX = (X(NX)-X(NX-1))/(X(NX-1)-X(NX-2))
      A(NX,1) = (1.+GRX)*A(NX-1,1) - GRX*A(NX-2,1)
      RHOVX(NX,NZ) = RHOVX0(NX)
      RHOVZ(NX,NZ) = RHOVZ0(NX)
      U(NX,NZ) = U0(NX)
      RHO(NX,NZ) = RHO0(NX)
      BX(NX,NZ) = BX0(NX)
      BZ(NX,NZ) = BZ0(NX)
      A(NX,NZ) = A0(NX)
      RETURN
  600 CONTINUE
C
C         ENTRY FOR IMPOSING BOUNDARY CONDITIONS ON TRANSPORT VELOCITY
C         ENTRY 601 FOR X-SCAN, VX(I),I = 1,NX,
C         ENTRY 606 FOR Z-SCAN, VZ(J),J = 1,NZ.
C
      IF(LTEST)WRITE(10,9996)
```

```
      GOTO (601,606) , JRZ
  601 CONTINUE
      VELOCH(1) = -VELOCH(3)
      VELOCH(NX) = VELOCH(NX-2)
      RETURN
  606 CONTINUE
      VELOCH(1) = -VELOCH(3)
      VELOCH(NZ) = RHOVZ0(JRZ)/RHO0(JRZ)
      RETURN
 9991 FORMAT(16H         BOUNDY 1 )
 9993 FORMAT(16H         BOUNDY 3 )
 9994 FORMAT(16H         BOUNDY 4 )
 9995 FORMAT(16H         BOUNDY 5 )
 9996 FORMAT(16H         BOUNDY 6 )
      END

      SUBROUTINE INITAL
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHO0(104),OU0(104),NX,NZ,DX,DZ,D
      COMMON/ISOTHM/TEMP
      COMMON/PARAM/WIDTH,BETA,RAP,DELON
      COMMON/BFLD0/BZ0(104),BX0(104),A0(104)
      BETA = 0.10
      TIME = 0.0
      PI = 3.141
      TEMP = BETA/2.
      W = 0.10
      GAMMA = 5.0/3.0
      DO 101 J = 1,NZ
      DO 101 I = 1,NX
      RHOVX(I,J) = 0.0
      BZ(I,J) = 0.0
      IF(Z(J).LE.W)BX(I,J) = 1.0*SIN(0.5*PI*Z(J)/W)
      IF(Z(J).GT.W)BX(I,J) = 1.0
      BX(I,3) = -BX(I,1)
      BX(I,2) = 0.0
      U(I,J) = 0.5*(1.0 + BETA - BX(I,J)*BX(I,J))
      RHO(I,J) = 1.0
      RHOVZ(I,J) = 0.0
  101 CONTINUE
      DO 102 I = 1,NX
      CALL VTMSTP
      CALL AFROMB
      DO 10 I = 1,NX
   10 A0(I) = A(I,NZ)
      RETURN
      END
      SUBROUTINE BFROMA
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
```

```
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      DO 10 I = 1,NX
      DO 10 J = 2,NZ-1
   10 BX(I,J) = -1.0*(A(I,J+1)-A(I,J-1))/(Z(J+1)-Z(J-1))
      DO 20 I = 2,NX-1
      DO 20 J = 1,NZ
   20 BZ(I,J) = (A(I+1,J)-A(I-1,J))/(X(I+1)-X(I-1))
C         USE B.C.'S TO COMPLETE CONVERSION
      DO 30 I = 1,NX
      BX(I,NZ) = 1.0/COS((REAL(I)*PI)/(REAL(NX)*10.0))
   30 BX(I,1) = -BX(I,3)
      DO 40 J = 1,NZ
      BZ(1,J) = -BZ(3,J)
   40 BZ(NX,J) = BZ(NX-2,J)
      RETURN
      END
      SUBROUTINE AFROMB
      COMMON/TIMING/DT,DTMIN,DTMAX,AK0,AK1,AK2,AK3
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      DIMENSION AO(104)
      REAL GAUGE
      PI = 3.141
      if(time.le.2.0)vinp = 0.05*(time/2.0)
      if(time.ge.2.0)vinp = 0.05
      if(time.le.2.)factor = time/2.
      if(time.ge.2.)factor = 1.0
      DECR = 1.0
      DO 10 I = 1,NX
      AO(I)=-1.0-DECR*VINP*TIME-((0.03+(factor*0.07))/(0.5*PI))*
     2COS((REAL(I)*PI)/(REAL(NX)*2.0))
      A(I,NZ) = AO(I)
      A(I,NZ-2) = A(I,NZ) + BX(I,NZ)*(Z(NZ)-Z(NZ-2))
      GRZ = (Z(NZ-1)-Z(NZ-2))/(Z(NZ)-Z(NZ-2))
      A(I,NZ-1) = A(I,NZ-2)*GRZ + (1.0-GRZ)*A(I,NZ)
10    CONTINUE
C         COMPUTE A ARRAY FROM BX AND BZ
      DO 30 I = 1,NX
      DO 30 J = NZ-2,3,-1
30    A(I,J-1) = A(I,J+1) + 1.0*BX(I,J)*(Z(J+1)-Z(J-1))
      DO 31 I = 1,NX
31    A(I,1) = A(I,3)
      RETURN
      END
```

Part Three: Boundary and initial conditions for the tearing mode problem. Two sets of boundary conditions are listed, showing the numerical implementation of the open-ended and symmetrical tearing cases.

```
      SUBROUTINE BOUNDY(NO,JRZ)
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/MHDRZI/RHOI(104),RHOVXI(104),RHOVZI(104),BXI(104),BZI(104),
     2              UII   (104),
     3              RHOH(104),RHOVXH(104),RHOVZH(104),BXH(104),BZH(104),
     4              UHH   (104)
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB,
     1              SWOHMH
      COMMON/TIMING/DT,DTMIN,DTMAX,AK0,AK1,AK2,AK3
      COMMON/SYSCON/RSC,ANISOT,ZBDIF,ZKDIF,CPLRAD,PSI
      DIMENSION UO(104),RHOO(104),RHOVXO(104),AO(104),BXO(104),BZO(104)
     1,RHOVZO(104),DIVV(104),LINE(104)
      GO TO (500,500,300,400,500,600),NO
  300 CONTINUE
C
C         ENTRY AFTER HALF TIME STEP IN X-CONVECTION
C
      IF(LTEST) WRITE(10,9993)
      BZH(NX) = BZH(NX-2)
      BZH(1) = -BZH(3)
      BXH(NX) = BXH(NX-2)
      BXH(1) = BXH(3)
      RHOH(NX) = RHOH(NX-2)
      RHOH(1) = RHOH(3)
      RHOVXH(NX) = RHOVXH(NX-2)
      RHOVXH(1) = -RHOVXH(3)
      RHOVZH(NX) = RHOVZH(NX-2)
      RHOVZH(1) = RHOVZH(3)
      UHH(NX) = UHH(NX-2)
      UHH(1) = UHH(3)
      RETURN
  400 CONTINUE
C
C         ENTRY AFTER HALF TIME STEP IN Z-CONVECTION
C
      IF(LTEST)WRITE(10,9994)
      BZH(NZ) = BZH(NZ-2)
      BZH(1) = BZH(3)
      BXH(NZ) = BXH(NZ-2)
      RHOVXH(NZ) = RHOVXH(NZ-2)
      RHOVZH(NZ) = RHOVZH(NZ-2)
      UHH(NZ) = UHH(NZ-2)
      RHOH(NZ) = RHOH(NZ-2)
```

```
      BXH(1) = -BXH(3)
      RHOH(1) = RHOH(3)
      RHOVXH(1) = RHOVXH(3)
      RHOVZH(1) = -RHOVZH(3)
      UHH(1) = UHH(3)
      RETURN
  500 CONTINUE
C
C         ENTRY AFTER FULL TIME STEP IN DIFFUSION.
C         BOUNDARIES MUST BE MADE CONSISTENT FOR INPUT TO CONVECTION.
C
      IF(LTEST) WRITE(10,9995)
C  ---LEFT HAND BOUNDARY----

      DO 110 J = 2,NZ-1

         BZ(1,J) = -BZ(3,J)
         BZ(2,J) = 0.0
         RHOVX(1,J) = -RHOVX(3,J)
         RHOVX(2,J) = 0.0
         BX(1,J) = BX(3,J)
         RHOVZ(1,J) = RHOVZ(3,J)
         U(1,J) = U(3,J)
         RHO(1,J) = RHO(3,J)
         A(1,J) = A(3,J)
110   CONTINUE

C  ----TOP BOUNDARY----

      DO 120 I = 1,NX-1
         BX(I,1) = -BX(I,3)
         BX(I,2) = 0.0
         RHOVZ(I,1) = -RHOVZ(I,3)
         RHOVZ(I,2) = 0.0
         BZ(I,1) = BZ(I,3)
         RHOVX(I,1) = RHOVX(I,3)
         U(I,1) = U(I,3)
         RHO(I,1) = RHO(I,3)
         A(I,1) = A(I,3)
120   CONTINUE

C  ----RIGHT HAND BOUNDARY----

      DO 130 J = 2,NZ-1
         BZ(NX,J) = BZ(NX-2,J)
         RHOVX(NX,J) = RHOVX(NX-2,J)
         RHOVZ(NX,J) = RHOVZ(NX-2,J)
         U(NX,J) = U(NX-2,J)
         RHO(NX,J) = RHO(NX-2,J)
         G1 = (X(NX)-X(NX-2))/(Z(J+1)-Z(J-1))
         BX(NX,J) = BX(NX-2,J) - G1*(BZ(NX-1,J+1)-BZ(NX-1,J-1))
         A(NX,J) = A(NX-2,J) + 1.0*BZ(NX-1,J)*(X(NX)-X(NX-2))
130   CONTINUE

C  ----UPPER RIGHT HAND CORNER CONDITION----
```

```
      RHOVX(NX,1) = RHOVX(NX-2,1)
      RHOVZ(NX,1) = RHOVX(NX-2,1)
      RHO(NX,1) = RHO(NX-2,1)
      U(NX,1) = U(NX-2,1)
      BX(NX,1) = -BX(NX,3)
      BX(NX,2) = 0.0
      BZ(NX,1) = BZ(NX,3)
      G = (X(NX)-X(NX-1))/(X(NX-1)-X(NX-2))
      A(NX,1) = (1.+G)*A(NX-1,1) - G*A(NX-2,1)

C  ----BOTTOM BOUNDARY----

      DO 140 I = 2,NX-1
         BX(I,NZ) = BX(I,NZ-2)
         RHOVX(I,NZ) = RHOVX(I,NZ-2)
         RHOVZ(I,NZ) = RHOVZ(I,NZ-2)
         U(I,NZ) = U(I,NZ-2)
         RHO(I,NZ) = RHO(I,NZ-2)
         G1 = (Z(NZ)-Z(NZ-2))/(X(I+1)-X(I-1))
         BZ(I,NZ) = BZ(I,NZ-2) - G1*(BX(I+1,NZ-1)-BX(I-1,NZ-1))
         G2 = (Z(NZ)-Z(NZ-1))/(Z(NZ-1)-Z(NZ-2))
         A(I,NZ) = A(I,NZ-2) - 1.0*BX(I,NZ-1)*(Z(NZ)-Z(NZ-2))
140   CONTINUE

C  ----LOWER LEFT HAND CORNER CONDITION----

      RHOVX(1,NZ) = RHOVZ(1,NZ-2)
      RHOVZ(1,NZ) = RHOVZ(1,NZ-2)
      U(1,NZ) = U(1,NZ-2)
      RHO(1,NZ) = RHO(1,NZ-2)
      BZ(1,NZ) = -BZ(3,NZ)
      BZ(2,NZ) = 0.0
      BX(1,NZ) = BX(3,NZ)
      G = (Z(NZ)-Z(NZ-1))/(Z(NZ-1)-Z(NZ-2))
      A(1,NZ) = (1.+G)*A(1,NZ-1) - G*A(1,NZ-2)

C  ----LOWER RIGHT HAND CORNER CONDITION----

      BZ(NX,NZ) = BZ(NX-2,NZ)
      BX(NX,NZ) = BX(NX,NZ-2)
      G1 = (Z(NZ)-Z(NZ-2))/(X(NX)-X(NX-2))
      BZ(NX-1,NZ) = BZ(NX-1,NZ-2) - G1*(BX(NX,NZ-1)-BX(NX-2,NZ-1))
      G2 = (X(NX)-X(NX-2))/(Z(NZ-2)-Z(NZ))
      BX(NX,NZ-1) = BX(NX-2,NZ-1) - G2*(BZ(NX-1,NZ-2)-BZ(NX-1,NZ))
      RHOVX(NX,NZ) = RHOVX(NX-2,NZ)
      RHOVZ(NX,NZ) = RHOVZ(NX-2,NZ)
      U(NX,NZ) = U(NX-2,NZ)
      RHO(NX,NZ) = RHO(NX-2,NZ)
      A(NX,NZ) = A(NX-2,NZ) + BZ(NX-1,NZ)*(X(NX)-X(NX-2))
      A(NX-1,NZ) = (A(NX,NZ) + A(NX-2,NZ))/2.
      RETURN
  600 CONTINUE
C
C         ENTRY FOR IMPOSING BOUNDARY CONDITIONS ON TRANSPORT VELOCITY
```

```
C         ENTRY 601 FOR X-SCAN, VX(I),I = 1,NX,
C         ENTRY 606 FOR Z-SCAN, VZ(J),J = 1,NZ.
C
      IF(LTEST)WRITE(10,9996)
      GOTO (601,606) , JRZ
  601 CONTINUE
      VELOCH(1) = -VELOCH(3)
      VELOCH(NX) = VELOCH(NX-2)
      RETURN
  606 CONTINUE
      VELOCH(1) = -VELOCH(3)
      VELOCH(NZ) = VELOCH(NZ-2)
      RETURN
 9991 FORMAT(16H         BOUNDY 1 )
 9993 FORMAT(16H         BOUNDY 3 )
 9994 FORMAT(16H         BOUNDY 4 )
 9995 FORMAT(16H         BOUNDY 5 )
 9996 FORMAT(16H         BOUNDY 6 )
      END
```

```
      SUBROUTINE BOUNDY(NO,JRZ)
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/MHDRZI/RHOI(104),RHOVXI(104),RHOVZI(104),BXI(104),BZI(104),
     2              UII   (104),
     3              RHOH(104),RHOVXH(104),RHOVZH(104),BXH(104),BZH(104),
     4              UHH   (104)
      COMMON/TESTER/ LTEST,LPRINT,LINIT
      COMMON/CONSRC/SGRO(104),SSRO(104),SHRO(104),SQRO(104),
     2              SGVR(104),SSVR(104),SHVR(104),SQVR(104),
     3              SGVZ(104),SSVZ(104),SHVZ(104),SQVZ(104),
     4              SGBR(104),SSBR(104),SHBR(104),SQBR(104),
     5              SGBZ(104),SSBZ(104),SHBZ(104),SQBZ(104),
     6              SGEQ(104),SSEQ(104),SHEQ(104),SQEQ(104),
     7              VELOCH(104),ZBCX,BCVRNZ,BCVZ1Z,BCBX1Z,BCBXNZ
      COMMON/SWITCH/SWGRAV,SWTEMP,SWRADN,SWTDIF,SWBDIF,SW2DIM,SWMHDB,
     1              SWOHMH
      COMMON/TIMING/DT,DTMIN,DTMAX,AKO,AK1,AK2,AK3
      COMMON/SYSCON/RSC,ANISOT,ZBDIF,ZKDIF,CPLRAD,PSI
      DIMENSION UO(104),RHOO(104),RHOVXO(104),AO(104),BXO(104),BZO(104)
     1,RHOVZO(104),DIVV(104),LINE(104)
      GO TO (500,500,300,400,500,600),NO
  300 CONTINUE
C
C         ENTRY AFTER HALF TIME STEP IN X-CONVECTION
C
      IF(LTEST) WRITE(10,9993)
      BZH(NX) = -BZH(NX-2)
      BZH(1) = -BZH(3)
      BXH(NX) = BXH(NX-2)
      BXH(1) = BXH(3)
      RHOH(NX) = RHOH(NX-2)
      RHOH(1) = RHOH(3)
      RHOVXH(NX) = -RHOVXH(NX-2)
      RHOVXH(1) = -RHOVXH(3)
      RHOVZH(NX) = RHOVZH(NX-2)
      RHOVZH(1) = RHOVZH(3)
      UHH(NX) = UHH(NX-2)
      UHH(1) = UHH(3)
      RETURN
  400 CONTINUE
C
C         ENTRY AFTER HALF TIME STEP IN Z-CONVECTION
C
      IF(LTEST)WRITE(10,9994)
      BZH(NZ) = BZH(NZ-2)
      BZH(1) = BZH(3)
      BXH(NZ) = BXH(NZ-2)
      RHOVXH(NZ) = RHOVXH(NZ-2)
      RHOVZH(NZ) = RHOVZH(NZ-2)
      UHH(NZ) = UHH(NZ-2)
      RHOH(NZ) = RHOH(NZ-2)
```

```
      BXH(1) = -BXH(3)
      RHOH(1) = RHOH(3)
      RHOVXH(1) = RHOVXH(3)
      RHOVZH(1) = -RHOVZH(3)
      UHH(1) = UHH(3)
      RETURN
  500 CONTINUE
C
C         ENTRY AFTER FULL TIME STEP IN DIFFUSION.
C         BOUNDARIES MUST BE MADE CONSISTENT FOR INPUT TO CONVECTION.
C
      IF(LTEST) WRITE(10,9995)
C  ---LEFT HAND BOUNDARY----

      DO 110 J = 2,NZ-1

         BZ(1,J) = -BZ(3,J)
         BZ(2,J) = 0.0
         RHOVX(1,J) = -RHOVX(3,J)
         RHOVX(2,J) = 0.0
         BX(1,J) = BX(3,J)
         RHOVZ(1,J) = RHOVZ(3,J)
         U(1,J) = U(3,J)
         RHO(1,J) = RHO(3,J)
         A(1,J) = A(3,J)
110   CONTINUE

C  ----TOP BOUNDARY----

      DO 120 I = 1,NX
         BX(I,1) = -BX(I,3)
         BX(I,2) = 0.0
         RHOVZ(I,1) = -RHOVZ(I,3)
         RHOVZ(I,2) = 0.0
         BZ(I,1) = BZ(I,3)
         RHOVX(I,1) = RHOVX(I,3)
         U(I,1) = U(I,3)
         RHO(I,1) = RHO(I,3)
         A(I,1) = A(I,3)
120   CONTINUE

C  ----RIGHT HAND BOUNDARY----

      DO 130 J = 2,NZ-1
      BZ(NX,J) = -BZ(NX-2,J)
      BZ(NX-1,J) = 0.0
      RHOVX(NX,J) = -RHOVX(NX-2,J)
      RHOVX(NX-1,J) = 0.0
      BX(NX,J) = BX(NX-2,J)
      RHOVZ(NX,J) = RHOVZ(NX-2,J)
      U(NX,J) = U(NX-2,J)
      RHO(NX,J) = RHO(NX-2,J)
      A(NX,J) = A(NX-2,J)
130   CONTINUE
```

```
C  ----BOTTOM BOUNDARY----

      DO 140 I = 2,NX-1
         BX(I,NZ) = BX(I,NZ-2)
         RHOVX(I,NZ) = RHOVX(I,NZ-2)
         RHOVZ(I,NZ) = RHOVZ(I,NZ-2)
         U(I,NZ) = U(I,NZ-2)
         RHO(I,NZ) = RHO(I,NZ-2)
         G1 = (Z(NZ)-Z(NZ-2))/(X(I+1)-X(I-1))
         BZ(I,NZ) = BZ(I,NZ-2) - G1*(BX(I+1,NZ-1)-BX(I-1,NZ-1))
         G2 = (Z(NZ)-Z(NZ-1))/(Z(NZ-1)-Z(NZ-2))
         A(I,NZ) = A(I,NZ-2) - 1.0*BX(I,NZ-1)*(Z(NZ)-Z(NZ-2))
140   CONTINUE

C  ----LOWER LEFT HAND CORNER CONDITION----

      RHOVX(1,NZ) = RHOVX(1,NZ-2)
      RHOVZ(1,NZ) = RHOVZ(1,NZ-2)
      U(1,NZ) = U(1,NZ-2)
      RHO(1,NZ) = RHO(1,NZ-2)
      BZ(1,NZ) = -BZ(3,NZ)
      BZ(2,NZ) = 0.0
      BX(1,NZ) = BX(3,NZ)
      G = (Z(NZ)-Z(NZ-1))/(Z(NZ-1)-Z(NZ-2))
      A(1,NZ) = (1.+G)*A(1,NZ-1) - G*A(1,NZ-2)

C  ----LOWER RIGHT HAND CORNER CONDITION----

      rhovx(nx,nz) = rhovx(nx,nz-2)
      rhovz(nx,nz) = rhovz(nx,nz-2)
      u(nx,nz) = u(nx,nz-2)
      rho(nx,nz) = rho(nx,nz-2)
      bz(nx,nz) = bz(nx,nz-2)
      bz(nx-1,nz) = 0.0
      a(nx,nz) = a(nx-2,nz)
      RETURN
  600 CONTINUE
C
C         ENTRY FOR IMPOSING BOUNDARY CONDITIONS ON TRANSPORT VELOCITY
C         ENTRY 601 FOR X-SCAN, VX(I),I = 1,NX,
C         ENTRY 606 FOR Z-SCAN, VZ(J),J = 1,NZ.
C
      IF(LTEST)WRITE(10,9996)
      GOTO (601,606) , JRZ
  601 CONTINUE
      VELOCH(1) = -VELOCH(3)
      VELOCH(NX) = -VELOCH(NX-2)
      RETURN
  606 CONTINUE
      VELOCH(1) = -VELOCH(3)
      VELOCH(NZ) = VELOCH(NZ-2)
      RETURN
 9991 FORMAT(16H        BOUNDY 1 )
 9993 FORMAT(16H        BOUNDY 3 )
```

```
9994 FORMAT(16H         BOUNDY 4 )
9995 FORMAT(16H         BOUNDY 5 )
9996 FORMAT(16H         BOUNDY 6 )
     END
```

```
      SUBROUTINE INITAL
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,nothere,VINP
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      COMMON/ISOTHM/TEMP
      COMMON/PARAM/WIDTH,BETA,RAP,DELON
      COMMON/BFLDO/BZO(104),BXO(104),AO(104)

      BETA = 0.10
      TIME = 0.0
      PI = 3.14159265358979
      TEMP = BETA/2.
      w = 0.05
      K = 123543
      GAMMA = 5.0/3.0
123   FORMAT (E10.3)
      DO 101 J = 1,NZ
      DO 101 I = 1,NX
      rhovx(i,j) = 0.0
      rhovz(i,j) = 0.0
      BZ(I,J) = EXP(-20.*Z(J))*0.01*(1.-RAN(K))
      BX(I,J) = TANH(Z(J)/W)
      BX(I,3) = -BX(I,1)
      BX(I,2) = 0.0
      U(I,J) = 0.5*(1.0 + BETA - BX(I,J)*BX(I,J))
      RHO(I,J) = 1.0
  101 CONTINUE
      DO 102 I = 1,NX
      ORHOO(I) = RHO(I,NZ)
      BXO(I) = BX(I,NZ)
  102 BZO(I) = BZ(I,NZ)
      CALL VTMSTP
      CALL AFROMB
      DO 10 I = 1,NX
   10 AO(I) = A(I,NZ)
      RETURN
      END
      SUBROUTINE BFROMA
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      DO 10 I = 1,NX
      DO 10 J = 2,NZ-1
   10 BX(I,J) = -1.0*(A(I,J+1)-A(I,J-1))/(Z(J+1)-Z(J-1))
      DO 20 I = 2,NX-1
      DO 20 J = 1,NZ
   20 BZ(I,J) = (A(I+1,J)-A(I-1,J))/(X(I+1)-X(I-1))
C         USE B.C.'S TO COMPLETE CONVERSION
      DO 30 I = 1,NX
      BX(I,NZ) = BX(I,NZ-2)
   30 BX(I,1) = -BX(I,3)
```

```
      DO 40 J = 1,NZ
      BZ(1,J) = -BZ(3,J)
   40 bz(nx,j) = -bz(nx-2,j)
      RETURN
      END
      SUBROUTINE AFROMB
      COMMON/TIMING/DT,DTMIN,DTMAX,AK0,AK1,AK2,AK3
      COMMON/FYSVAR/ RHO  (40,70),RHOVX(40,70),RHOVZ(40,70),
     2               BX   (40,70),BZ   (40,70),U    (40,70),
     3               ES   (40,70),A    (40,70),TIME,MEMDUM,
     4               REYN,BETA,VINP
      COMMON/METRIC/ X(104),Z(104),ORHOO(104),OUO(104),NX,NZ,DX,DZ,D
      DIMENSION AO(104)
      REAL GAUGE,DECR,GRX
      PI = 3.141
      DECR = 1.0
C       COMPUTE A ARRAY FROM BX AND BZ
      A(NX,1) = 0.0
      A(NX-2,1) = A(NX,1) - BZ(NX-1,1)*(X(NX)-X(NX-2))
C       APPLY B.C.
      GRX = (X(NX)-X(NX-1))/(X(NX-1)-X(NX-2))
      A(NX-1,1) = (A(NX,1)+GRX*A(NX-2,1))/(1.+GRX)
      DO 10 IR = 3,NX-1
      I = NX-IR+1
10    A(I-1,1) = A(I+1,1) - BZ(I,1)*(X(I+1)-X(I-1))
      A(NX,3) = -1.0*BX(NX,2)*(Z(3)-Z(1)) + A(NX,1)
C       APPLY B.C.
      GRZ = (Z(1)-Z(2))/(Z(2)-Z(3))
      A(NX,2) = (A(NX,1)+GRZ*A(NX,3))/(1.+GRZ)
      A(NX-2,2) = A(NX,2) - 1.0*BZ(NX-1,2)*(X(NX)-X(NX-2))
C       APPLY B.C.
      A(NX-1,2) = (A(NX,2)+GRZ*A(NX-2,2))/(1.+GRX)
      DO 20 IR = 2,NX-1
      I = NX-IR+1
20    A(I-1,2) = A(I+1,2) - 1.0*BZ(I,2)*(X(I+1)-X(I-1))
      DO 30 I = 1,NX
      DO 30 J = 2,NZ-1
30    A(I,J+1) = A(I,J-1) - 1.0*BX(I,J)*(Z(J+1)-Z(J-1))
      DO 60 I = 1,NX
60    A(I,1) = A(I,3)

      RETURN
      END
```

REFERENCES

Altyntsev, A.T., Lebedev, N.V., Krasov, V.I., Papernyi, V.L., Pis'ma Zh. Eksp. Teor. Fiz., 42, 360, 1985

Arnett, W.D., Commun. ACM, 28, 354, 1985

Bagenal, F., in Solar System Magnetic Fields, Reidel, Dordrecht, 1985

Bateman, G., MHD Instabilities, Cambridge, Mass.: MIT Press, 1978

Bayliss, A., Turkel, E., J. Comp. Phys., 48, 182, 1982

Bernstein, I.B., Frieman, E.A., Kruskal, M.D., Kulsrud, R.M., Proc. Roy. Soc. (A), 244, 17, 1958

Bhattacharjee, A., Tajima, T., Brunel, F., Phys. Fluids, 26, 3332, 1983

Birn, J., Computer simulation of reconnection in planetary magnetospheres, in Unstable current systems and plasma instabilities in astrophysics, IAU Symp. No. 107, Reidel, Dordrecht, 1985

Biskamp, D., Welter, H., Phys. Rev. Lett., 44, 1069, 1980

Biskamp, D., in Magnetic Reconnection and Turbulence, Les Editions de Physique, Les Ulis, France, 1985

Biskamp, D., Phys Fluids, 29, 1520, 1986

Boris, J.P., Book, D.L., J. Comp. Phys., 11, 38, 1973

Boris, J.P., Book, D.L., J. Comp. Phys., 20, 397, 1976

Burden, R.L., Faires, J.D., Reynolds, A.L., Numerical analysis, Prindle, Weber, Schmidt, 1981

Camerlengo, A.L., O'Brien, J.J., J. Comp. Phys., 35, 12, 1980

Chen, F.F., Introduction to Plasma Physics, Plenum, 1974

Chu, C.K., Sereney, A., J. Comp. Phys., 15, 476, 1974

Courant and Hilbert, Methods of Mathematical Physics, vol. 2, Interscience, 1962

Eastwood, J.W., Arter, W., Culham Rep., CLM - P763, 1985

Eddy, J.A., A New Sun: The Solar Results from Skylab, NASA S.P. 402, U.S. Government Printing Office, Washington D.C., 1979

Finn, J.M., Kaw, P.K., Phys Fluids, 20, 72, 1977

Forbes, T.G., Priest, E.R., Solar Phys., 81, 303, 1982

Forbes, T.G., Priest, E.R., J. Geophys. Res., 88, 863, 1983

Forbes, T.G., Priest, E.R., Solar Phys., 84, 169, 1983a

Forbes, T.G., Priest, E.R., Solar Phys., 88, 211, 1983b

Forbes, T.G., Priest, E.R., Solar Phys., 94, 315, 1984

Forbes, T.G., Priest, E.R., submitted to Rev. Geophys., 1987

Fukao, S., Tsuda, T., Planet. Space Sci., 21, 1151, 1973

Furth, H.P., Phys. Fluids, 28, 1595, 1985

Furth, H.P., Killeen, J., Rosenbluth, M.N., Phys. Fluids, 6, 459, 1963

Grad, H., Phys. Fluids, 7, 1283, 1964

Han, T.Y., Meng, J.C.S., Innis, G.E., J. Comp. Phys., 49, 276, 1983

Hayashi, T., Jour. Phys. Soc. Japan, 50, 3124, 1981

Hayashi, T., Sato, T., Jour. Geophys. Res., 83, 217, 1978

Hu, Y.Q., Wu, S.T., J. Comp. Phys., 55, 33, 1984

Israeli, M., Orszag, S.A., J. Comp. Phys., 41, 115, 1981

Landau, L.D., Lifschitz, E.M., Fluid Mechanics, Pergamon Press, 1959

Lee, L.C., Fu, Z.F., J. Geophys. Res., 91, 4551, 1986

Matthaeus, W.H., Montgomery, D., J. Plasma Phys., 25, 11, 1981

Mitchell, A.R., Computational methods in partial differential equations, John Wiley and Sons, 1969

Nakajima, H., Tajima, T., Brunel, F., Sakai, J., Bulletin of Faculty of Engineering, Toyama University, 1985

Nayfeh, A.H., Introduction to Perturbation Techniques, Wiley-Interscience, 1981

Newcomb, W.A., Annals of Physics, 10, 232, 1960

Nishida, A., in Magnetic Reconnection in Space and Laboratory Plasmas, Geophysical Monograph 30, American Geophysical Union, Washington D.C., 1984

Orlanski, I., J. Comp. Phys., 21, 251, 1976

Parker, E.N., Astrophys. J. Supl. Ser., 8, 177, 1963

Pearson, R.B., Richardson, J.L., Toussaint, D., Commun. ACM, 28, 385, 1985

Pellat, R., Sov. J. Plas. Phys., 9, 124, 1983

Petschek, H.E., in Proc. AAS-NASA Symp. on the Physics of Solar Flares, NASA S.P. 50, U.S. Government Printing Office, Washington D.C., 1964

Potter, D.E., in Plasma Physics and Nuclear Fusion Research, Academic Press, 1981

Priest, E.R., in Solar Magnetohydrodynamics, Reidel, Dordrecht, 1982

Priest, E.R., Rep. Prog. Phys., 48, 955, 1985

Priest, E.R., Forbes, T.G., J. Geophys. Res., 91, 5579, 1986

Pritchett, P.L., Wu, C.C., Phys. Fluids, 22, 2140, 1979

Roache, P.J., Computational fluid dynamics, Hermosa, Albuquerque, New Mexico, 1976

Robertson, J.A., Frank, J., Mon. Not. R. Astr. Soc., 221, 279, 1986

Robertson, J.A, Priest, E.R., submitted to Geophys. Astrophys. Fluid Dyn., 1987

Sakai, J., J. Plasma Phys., 30, 109, 1983

Sato, T., Hayashi, T., Phys. Fluids, 22, 1189, 1979

Schnack, D.D., PhD thesis, UCRL report 52399, 1978

Sonnerup, B.U.O., Priest, E.R., J. Plasma Phys., 14, 283, 1970

Soward, A.M., Priest, E.R., Phil. Trans. Roy. Soc. Lon., 284, 369, 1977

Soward, A.M., Priest, E.R., J. Plasma Physics, 35, 333, 1986

Steinolfson, R.S., Van Hoven, G., Phys. Fluids, 26, 117, 1983

Stenzel, R.L., in Magnetic Reconnection and Turbulence, Les Editions de Physique, Les Ulis, France, 1985

Sweet, P.A., IAU Symp. 6, 123, 1958

Syrovatsky, S.I., Sov. Phys. - JETP, 33, 933, 1971

Tsuda, T., Ugai, M., J. Plasma Phys., 18, 451, 1977

Ugai, M., Plasma Phys., 23, 857, 1981

Ugai, M., Phys. Fluids, 25, 1027, 1982

Ugai, M., Phys. Fluids, 26, 1569, 1983

Ugai, M., J. Plasma Phys., 26, 1549, 1984

Ugai, M., Tsuda, T., J. Plasma. Phys., 17, 337, 1977

Ugai, M., Tsuda, T., J. Plasma Phys., 21, 459, 1979a

Ugai, M., Tsuda, T., J. Plasma Phys., 22, 1, 1979b

Vasyliunas, V.M., Rev. Geophys, 13, 303, 1975

Weber, W.J., PhD thesis, The University of Utrecht, Utrecht, The Netherlands, 1978

Wesson, J.A., Nucl. Fusion, 6, 130, 1966

Wesson, J.A., Nucl. Fusion, 18, 87, 1978